Quantifying the spatio-temporal temperature dynamics of Greater London using thermal Earth observation

a thesis submitted by

Tomas du Chemin Holderness

for the degree of

Doctor of Philosophy

Newcastle University
School of Civil Engineering and Geosciences

October 2012

For Úna

Abstract

Urban areas are highly sensitive to extreme events such as heatwaves. In order to understand how cities will respond to thermal stress it is critical to quantify not only their temporal temperature dynamics but also their spatial temperature variability. However, many cities lack weather station networks with a sufficient spatial distribution to characterise spatio-temporal intra-urban temperature dynamics. One means by which spatially complete measurements of urban temperature may be derived is to employ satellite thermal Earth observed data. While some success has been achieved in understanding the temperature characteristics of cities using such data, relatively little work has been undertaken on establishing the use of long time-series Earth observed data as a supplement or alternative to screen-level air temperatures frequently utilised in urban climatological studies.

In this thesis a software framework, centred around the use of a spatial database, is developed which can be used to gain an improved understanding of how satellite thermal Earth observed data can be used in the long time-series analysis of urban temperature dynamics. The utility of the system is demonstrated by processing a 23 year time series (1985-2008) of 1,141 Advanced Very High Resolution Radiometer (AVHRR) images and hourly United Kingdom (UK) Met Office weather station measurements for the Greater London area. London was selected as the region of interest as it is the UK's only megacity, and has been shown to exhibit both a significant urban heat island and a severe increase in population mortality during previous heatwave events.

The software framework was employed to conduct two inter-related sets of analysis. First, the relationship over time between AVHRR estimated surface temperature (EST) and screen-level air temperature records is investigated and quantified. The resulting relationships are then used to produce an empirical model that can predict spatially complete summer-season air temper-

atures for London.

Cross-validation testing of the model at selected London weather stations showed model root mean square error (RMSE) ranging from 2.70 to 2.94°C and absolute errors in air temperature estimation of 0.45 to 1.67°C. A key finding of the thesis is that the minimal variation in prediction error between the different stations indicate a level of spatial robustness in the model across the urban surface, that is within the limits of the AVHRR EST precision. In addition, the model was used to estimate spatially averaged air temperatures over the Greater London area for selected summers, and showed a maximum error in air temperature prediction of 1.44°C. Furthermore, the prediction error for the heatwave summer of 2003 was 0.51°C, suggesting that such a model can successfully be used to estimate air temperatures for extreme heatwave summers. Such predictions are directly relevant to future assessments of urban population exposure to heatwaves, and it is envisaged that they could be used in conjunction with a population vulnerability index to create a spatially complete heatwave risk map for London.

This work is then extended to investigate the utility of satellite estimated surface temperature measurements to characterise temporally and spatially intra-urban heatwave dynamics using the commonly employed urban heat island intensity metric (UHII). Analysis of the AVHRR EST found that the data are highly sensitive to local meteorological conditions, and that temporal aggregation at the monthly scale is required to provide robust data-sets for inter-year analysis of summer temperatures and generation of the UHII metric. **S**tatistical testing of EST and air-temperature derived UHII for the heatwave summer of 2003 against other non-heatwave summers showed no significant increase in intensity at the 95% confidence level. This raises questions as to the applicability of the UHII metric to capture increases in urban temperatures during a heatwave event.

Acknowledgements

I would like to thank my supervisors Stuart Barr and Richard Dawson for their support and guidance throughout my PhD. In particular I am grateful to Stuart for his encouragement and enthusiasm at every stage of the project and to Richard who afforded a me number of opportunities to travel and engage with other researchers.

Acknowledgement is given to the Engineering and Physical Sciences Research Council and the School of Civil Engineering and Geosciences for funding the research, and who made it possible for me to attend several conferences and summer schools. I am also grateful to the Dundee Satellite Receiving Station and the Natural Environment Research Council for provision of the AVHRR data. A special mention must go to Andrew Brooks and Neil Lonie at Dundee for their assistance in answering my numerous technical questions.

Lastly, I would like to express my appreciation and thanks to my family, especially my parents Anne and Richard for their continued support and sound advice, and to Milly, whose endless optimism and encouragement were invaluable.

Table of Contents

List of Figures

List of Tables

List of abbreviations and notations

API	Application Programming Interface
AOT	Aerosol Optical Thickness
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
AVHRR	Advanced Very High Resolution Radiometer
BADC	British Atmospheric Data Centre
BNG	British National Grid
CRED	Centre for Research on the Epidemiology of Disasters
CSV	Comma Separated Value
DLR	Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Center)
ETM+	Enhanced Thematic Mapper Plus
ERS	European Remote Sensing Satellite
EST	Estimated Surface Temperature
GDAL	Geospatial Data Abstraction Library
GLA	Greater London Authority
GUI	Graphical User Interface
HHWS	Heat-Health Warning System
HTCI	Human Thermal Comfort Index
HTML	Hyper Text Markup Language
InSAR	Synthetic Aperture RADAR Interferometry
MIDAS	Met Office Integrated Data Archive System
MODIS	Moderate Resolution Imaging Spectroradiometer

NDVI	Normalised Difference Vegetation Index
NumPy	Numerical Python
QGIS	Quantum Geographic Information System
RADAR	Radio Detection And Ranging
RMSE	Root Mean Square Error
RTM	Radiative Transfer Model
SAR	Synthetic Aperture RADAR
SciPy	Scientific Python
SQL	Structured Query Language
SPOT	Sytème Pour l'Observation De La Terre
SWT	Split Window Technique
TM	Thematic Mapper
UBL	Urban Boundary Layer
UCL	Urban Canopy Layer
UHI	Urban Heat Island
UHII	Urban Heat Island Intensity
UKMO	United Kingdom Met Office
VHRR	Very High Resolution Radiometer
WGS	World Geodetic System
WKT	Well Known Text
XML	Extensible Markup Language
C	Celsius
K	Kelvin
$\overline{x}$	Mean
σ	Standard Deviation
λ	Wavelength
μm	Micrometer
$Wm^{-2}sr^{-1}$	Watts per Square Meter per Steradian

Chapter 1

Introduction

1.1 Extreme temperature events and urban climate modification

Heatwaves increase the morbidity and mortality of vulnerable populations such as the elderly, infirm and very young (Fish et al., 1985; Johnson et al., 2005; Kovats et al., 2006). In North America, and northern and western Europe heatwaves cause more deaths than other weather related disasters including hurricanes, floods and storm surges (CRED, 2012). The summer heatwave of 2003 was particularly severe and caused more than 30,000 excess deaths in Europe (Bhattacharya, 2003), with over 19,000 deaths in France, 9,000 in Germany and 2,000 in England and Wales (Kovats et al., 2006; CRED, 2012). The problem is compounded further by global climate change which has increased the severity and frequency of heatwave events over the last century (Hansen et al., 2012), and is predicted to continue to do so in the future (Rosenzweig et al., 2011).

Urban populations are especially at risk from heatwaves as exposure to the heat hazard is elevated by the urban heat island effect (Johnson et al., 2005; Dousset et al., 2011). The urban heat island is an inadvertent local-scale climate modification which causes city temperatures to be significantly warmer than the surrounding rural hinterland (Oke, 1987; Wilby, 2003), and is the

result of urban surface materials altering the surface-atmosphere energy balance which increases urban temperatures and reduces cooling (Oke, 1982, 1987).

Consequently, during heatwave periods the urban heat island can intensify temperatures (Johnson et al., 2005; Dousset et al., 2011; Rosenzweig et al., 2011), and there are a number of examples where this has resulted in increased morbidity and mortality rates in vulnerable populations within a city (Johnson et al., 2005; Dousset et al., 2011). During the 2003 summer heatwave there was a 17% increase in excess deaths in England, and a 1% increase in morbidity (as measured by hospital admissions compared to summer months over the five previous years) (Johnson et al., 2005). London was the most severely affected region and experienced a 42% increase in mortality and 6% increase in morbidity (Johnson et al., 2005). The marked increase in London is attributed to the extreme temperatures in the city, which were up to 5.7°C warmer compared to other regions, including the South East (Johnson et al., 2005).

As a result of such events a number of studies have cited the need to monitor and quantify urban temperature dynamics, including the urban heat island, over long-time series to better understand the response of cities to heatwave events (Harlan et al., 2006; Gaffin et al., 2008; Jones and Lister, 2009) for future mitigation and adaptation options (Solecki et al., 2005; Harlan et al., 2006; Dousset et al., 2011). Furthermore, it has been shown that the urban heat island is spatially variable (Roth et al., 1989; Harlan et al., 2006; Jenerette et al., 2007) and as a result the exposure to extreme temperature events within cities is also spatially variable (Harlan et al., 2006; Dousset et al., 2011). For example, Dousset et al. (2011) showed that during the 2003 summer heatwave a temperature difference of 0.5°C between neighbourhoods in Paris resulted in vulnerable populations in the warmer neighbourhood being twice as likely to experience heat related mortality. As such, a number of studies have cited the need to capture, monitor and quantify intra-urban temperature dynamics in a spatially complete manner (Eliasson and Svensson, 2003; Harlan et al., 2006; Gaffin et al., 2008) so that future heatwave

adaptation and mitigation options can account for spatial variability in temperatures during heatwave periods (Harlan et al., 2006; Dousset et al., 2011; Rosenzweig et al., 2011).

1.2 Quantifying intra-urban temperature dynamics

Traditionally, the urban heat island has been quantified using the urban heat island intensity metric (UHII) derived from near-surface (1-2m) terrestrial air temperature measurements from weather station networks (Oke, 1987; Wilby, 2003; Jones and Lister, 2009). The UHII is defined as the difference between urban and rural temperatures for a given point in time (Oke, 1987), and has been used extensively to define urban heat islands for cities around the world (Oke, 1987; Balling and Brazel, 1987; Gaffin et al., 2008; Jones and Lister, 2009; Cai et al., 2011). However, in many cities weather station networks are too spatially sparse for air temperature measurements to truly capture intra-urban temperature variability at the neighbourhood level (Eliasson and Svensson, 2003; Harlan et al., 2006). This has been remediated to some extent by studies using air temperature measurements from vehicle traverses and bespoke networks of spatially-dense temperature loggers to capture spatially variations in temperature and urban heat island intensity over a particular city (Oke, 1973; Eliasson and Svensson, 2003; Harlan et al., 2006). The disadvantage of such methods is that they can be costly to implement and, as such, data is often only available for limited time periods (Oke, 1973; Eliasson and Svensson, 2003; Kolokotroni and Giridharan, 2008).

A number of studies have demonstrated the use of thermal Earth observed surface temperatures as supplementary or alternative data to terrestrial air temperature measurements to quantify intra-urban temperature dynamics (Nichol, 1996; Roth et al., 1989; Voogt and Oke, 2003; Tran et al., 2006; To et al., 2011). The key advantage of thermal Earth observation in this regard is that it provides spatially-complete coverage of surface temperatures

(Roth et al., 1989; Voogt and Oke, 2003). Furthermore, many satellite thermal Earth observation sensors provide repeated measurements of the same location (Streutker, 2003; Cheval and Dumitrescu, 2009), and there are now archives of thermal imagery covering up to 30 years, meaning that such data has the potential to be used for long time-series, statistically robust, analysis of intra-urban temperature dynamics (Streutker, 2003; Cheval et al., 2009; Tomlinson et al., 2012). However, there are substantial technical challenges associated with the storage and manipulation of long time-series Earth observation data-sets due to its voluminous nature (Latifovic et al., 2005). In this regard, one of the biggest restrictions to the use of such data to quantify urban temperatures is the availability of software which can integrate terrestrial and thermal Earth observed measurements in a spatio-temporal manner, so that comparison and analysis of the two can take place for the same time and location within a city (Vogt et al., 1997; Nichol, 2009). As a result, relatively little work has been undertaken to examine the utility of long time-series thermal Earth observation to estimate near-surface temperatures and to capture significant changes in intra-urban temperatures during extreme temperature events in cities.

1.3 Aim and objectives

The aim of this study is to explore the utility of long time-series thermal Earth observation to quantify intra-urban temperature dynamics during heatwave events. To achieve this aim the following objectives will be addressed:

1. Identify and review existing methodologies for quantifying intra-urban temperature dynamics using both terrestrial and Earth observed measurements and critically assess their ability to capture changes in urban temperatures during heatwave events.

2. Develop a software framework to enable the spatio-temporal integration, processing and analysis of terrestrial air temperature measurements and thermal Earth observation data.

3. Examine the relationship between near-surface air temperature and Earth observed surface temperatures to evaluate the potential of deriving empirical models to estimate spatially complete air temperatures during heatwave events using thermal Earth observation.

4. Assess the ability of thermal Earth observation to capture, monitor and quantify changes in intra-urban temperatures and urban heat island intensity during heatwave events.

1.4 Thesis outline

The main aim and key objectives outlined in Section 1.3 are addressed in this thesis and presented in the following chapters. Chapter 2 presents a synopsis of the literature covering urban temperature dynamics and the urban heat island effect, as well as a detailed review on the use of thermal Earth observation to study the temperature regime of cities. Chapter 3 describes the data-sets selected for analysis and the methods used for pre-processing of the data, while Chapter 4 presents the software framework developed to provide a platform for integration and analysis of the data. An examination of the relationship between near-surface air temperature and thermal Earth observed surface temperatures, including the potential for empirical modelling of air temperatures is covered in Chapter 5. Chapter 6 evaluates the utility of thermal Earth observation to capture and quantify intra-urban temperature dynamics during a heatwave period. The results are analysed and discussed in Chapter 7, and final conclusions are drawn in Chapter 8.

Chapter 2

Thermal Earth observation and the urban heat hazard

2.1 Introduction

With over half the world's population living in urban areas (UN, 2009; Rosenzweig et al., 2011), cities are at the forefront of challenges posed by climate change (Rosenzweig et al., 2011). Global climate change will exert added stress on cities in the future through a greater frequency of extreme events such as heatwaves, droughts, floods and storm surges (IPCC, 2007; Wilby, 2007; Rosenzweig et al., 2011). Additionally, cities are concentrates of populations meaning that such events affect a large number of people (Barata et al., 2011) and who collectively by means of their actions and consumption patterns can induce local climate modifications, which in turn can increase the magnitude of heatwaves, floods and extreme rainfall (Shepherd et al., 2002; Wilby, 2007; Gaffin et al., 2008; Jones et al., 2008).

In North America and Northern and Western Europe, heatwaves cause more deaths than other extreme meteorological events such as flooding or storms (Rosenzweig et al., 2011; CRED, 2012). As such, future changes in heatwave intensity, frequency and impacts are currently the subject of research at international, national and city scales (McCarthy, 2001; Solecki et al., 2005; IPCC,

2007; Gaffin et al., 2008; Wilby, 2008; Mehrotra et al., 2009; Blake et al., 2011; Mehrotra et al., 2011; Rosenzweig, 2011; Rosenzweig et al., 2011). The influence of climate change on the frequency and intensity of extreme events is discussed further in Section 2.2.

Studies examining the effects of heatwaves in cities have also investigated the urban heat island effect; a local city-scale climate modification which was first recorded in the nineteenth century (Howard, 1833; Oke, 1973, 1987; Balling and Brazel, 1987; Lee, 1992; Cai et al., 2011). The urban heat island has been studied intensively with regards to its interaction with climate change and heatwaves (Solecki et al., 2005; Wilby, 2007; Gaffin et al., 2008; Blake et al., 2011; Mehrotra et al., 2011; Rosenzweig et al., 2011). A number of studies have shown that the urban heat island has increased exposure to heatwaves (Rosenzweig et al., 2005; Barata et al., 2011; Blake et al., 2011), although further research is required into heat island measurement to better assess future urban exposure to heatwave hazards (Gaffin et al., 2008; Blake et al., 2011). These interactions are examined in more detail in Section 2.3.2.

Quantification of the urban heat island has traditionally been conducted using terrestrial air temperature measurements (Oke, 1987; Eliasson and Svensson, 2003). However, remotely-sensed data has also been employed to record the temperature dynamics of urban areas since the 1970's (Roth et al., 1989; Gallo et al., 1993; Voogt and Oke, 2003) and is commonly used in urban heat island studies as an alternative or supplementary data for terrestrial temperature measurements (Gallo et al., 1993, 1995; Voogt and Oke, 2003; Dousset and Gourmelon, 2003; Cheval and Dumitrescu, 2009; Cai et al., 2011). This chapter presents a review of the literature most relevant to the aim of the thesis, first, to identify the relevant research challenges a review of urban temperature dynamics and the urban heat island effect is presented (Sections 2.2 & 2.3). Second, an extensive review is conducted on the use of Earth observation to study the temperature regime of cities (Section 2.4), to recognise the research challenges that need to be addressed in order for Earth observation to be robustly applied in studies investigating urban temperature dynamics.

2.2 Extreme Temperature Events and Cities

Heatwaves endanger the health of vulnerable populations such as the elderly, infirm and very young (Rosenzweig et al., 2011) and have been shown to cause higher rates of morbidity and mortality than other extreme weather events such as floods and storms world wide (Rosenzweig, 2011; Rosenzweig et al., 2011). Table 2.1 shows the top-10 climatological, meteorological and hydrological disasters (i.e. drought, extreme temperature (hot and cold), flood, mass (land) movement, storm and wildfire) by mortality in North America and Northern and Western Europe from 1900 to 2011. The table corresponds with the findings of Rosenzweig (2011); Rosenzweig et al. (2011), showing that heatwaves induce significantly more fatalities than any other type of disaster in these regions. The 2003 summer heatwave was particularly severe, with combined fatalities in France and Germany (28,845 deaths) being nearly 50% greater than all other disasters listed in the table (19,510 deaths).

Year	**Country**	**Region**	**Disaster Type**	**Number of Deaths**
2003	France	Paris	Heatwave	19490
2003	Germany	n/a	Heatwave	9355
1900	United States	Galveston (Texas)	Tropical cyclone	6000
1952	United Kingdom	London	Air pollution	4000
1953	Netherlands	Zuiderzee area	Storm surge/-coastal flood	2000
1928	United States	Lake Okeechobee (Florida)	Tropical cyclone	1836
2005	United States	Mobile, Bayou La Batre and more	Tropical cyclone	1833
2006	France	n/a	Heatwave	1388
1980	United States	Kansas City	Heatwave	1260
1936	United States	Illinois	Heatwave	1193

Table 2.1: Top 10 Climatological, meteorological and hydrological disasters by mortality for North America and Northern and Western Europe 1900-2011 (source: CRED (2012)).

As a result of climate change, heatwaves are predicted to grow in frequency and severity (IPCC, 2007; Rosenzweig et al., 2011; Hansen et al., 2012) and will be exacerbated by an increase in maximum and minimum urban temperatures (Gaffin et al., 2008; Rosenzweig et al., 2011). For example, under a high CO_2 emissions scenario, by the 2050's the average summer temperatures in the London region are expected to increase by 2.5-3.0°C (GLA, 2006). Furthermore, Hansen et al. (2012) demonstrate that globally over the past 30 years there has been a shift in the probability of the occurrence of extreme temperatures. The study shows that globally, heatwave events in the past 10 years (2001-2011) are up to five standard deviations hotter than average summer temperatures (as compared to a 1951-1980 baseline). Figure 2.1 shows a conceptual representation of the changing distribution of extreme temperature events as a result of climate change. Hansen et al. (2012) also found that the land surface area (urban and rural) over which such anomalies are occurring has increased from one to ten percent of the Earth's land surface in the same time period. The study concludes that warming since the 1980's has caused an increase in the intensity of extreme events and the distribution shift is so great that it can be argued that recent heatwaves (e.g. Texas 2011, Moscow 2010 and France 2003) would not have occurred without the influence of climate change (Hansen et al., 2012).

City populations are especially at risk from heatwaves due to inadvertent (i.e non-intentional) local climate modifications such as the urban heat island (UHI) effect. This increases urban temperatures and reduces nocturnal cooling (Oke, 1987) which in turn elevates population exposure and vulnerability to the heat hazard (Wilby, 2003; Rosenzweig et al., 2005; Solecki et al., 2005; GLA, 2006; Gaffin et al., 2008; Rosenzweig, 2011; Rosenzweig et al., 2011). In the urban climate change research network's (UCCRN) first assessment report on climate change and cities (ACR3), Rosenzweig et al. (2011) state that increasingly intense heatwaves along with exacerbated heat effects from the urban heat island are one of the key climate hazards facing cities. Figure 2.2 outlines how it is thought climate change determinants combined with urban modifying factors compound the impact of extreme events within cities

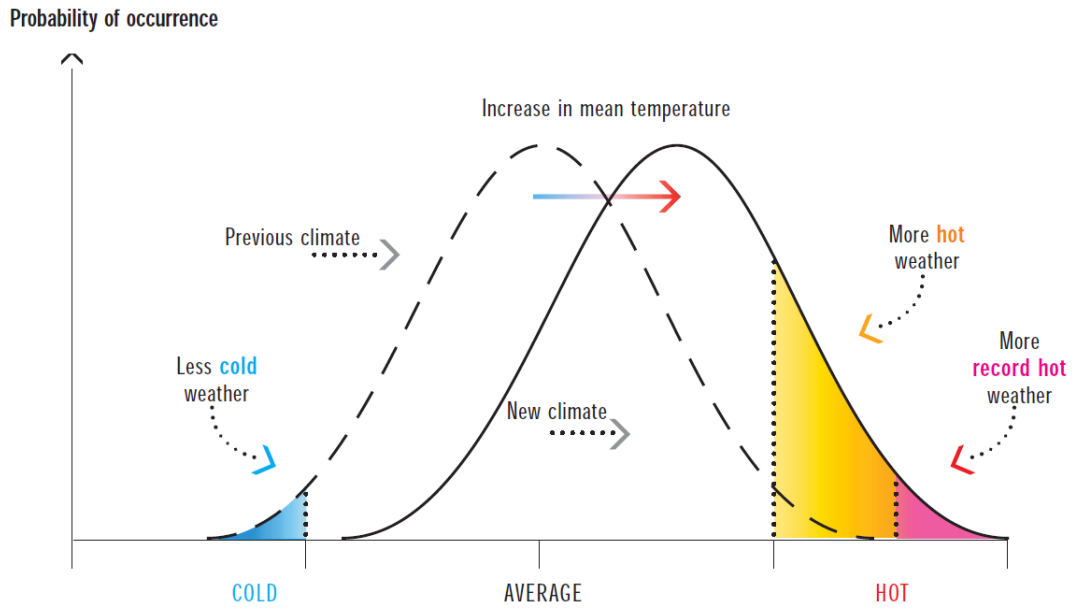


Figure 2.1: Conceptual diagram of the increasing probability of extreme temperature events as a result of global climate change (source: In Koppe et al. (2004) from Houghton et al. (2001)).

(Rosenzweig et al., 2011). For example, the exposure of the population to heat hazard during a heatwave can be increased by the urban heat island which elevates temperatures in urban areas (e.g. Dousset et al., 2011). Likewise, the exposure to a flood event is often greater in urban areas due to increased impervious surface cover (e.g. Booth and Leavitt, 1999). Such responses in the urban environment highlight the need to better understand the interactions between urban modification and extreme events to mitigate and adapt to resulting increases in exposure to the urban population.

The aforementioned urban heat island is one such urban modification, and is the result of a number of factors, primarily the elevated absorption and thermal re-radiation of solar energy by urban materials, and increases in anthropogenic heat sources (Oke, 1987; Betts and Best, 2004) (discussed further in Section 2.3). The impact of urban climate modification on heatwave exposure of urban populations can be seen in mortality statistics from previous events (Johnson et al., 2005; Kovats et al., 2006; Rosenzweig, 2011). For example during the heatwave summer of 2003, between the 4th and 13th August there were 2,091 excess deaths in the United Kingdom, 616 of which were in London (more than any other one area) with a further 447 deaths

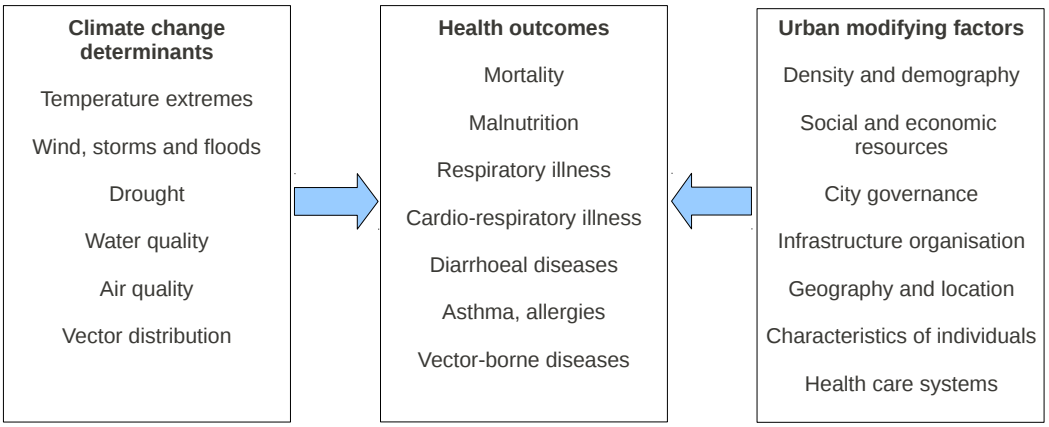


Figure 2.2: Climate change determinants and urban modifying factors on health outcomes in cities (source: Rosenzweig et al. 2011).

in the surrounding South East region (Johnson et al., 2005). The additional deaths in London were attributed to increased exposure to temperatures as a result of the urban heat island, which increased temperatures by up to 9°C in the city, compared to the surrounding rural area (see Section 2.3.2) (Johnson et al., 2005; GLA, 2006).

As a result of increased mortality and morbidity during previous heatwave events many countries are implementing heatwave forecasting and response plans at national, regional and city scales (Kovats and Hajat, 2008). Kovats and Ebi (2006) state that such systems are being implemented across Europe in the absence of sufficient evidence on the effectiveness of specific measures in reducing heatwave morbidity and mortality.

Heat-health warning systems (HHWS) link public health actions to meteorological forecasts of extreme temperature events (Kovats and Hajat, 2008; Hajat et al., 2010). A heat-health warning is issued if a weather forecast predicts temperatures greater than pre-defined threshold values (Hajat et al., 2010; Robinson, 2001). As population tolerance to heat varies depending on regional climate (Kovats and Ebi, 2006) there is no internationally accepted definition of a heatwave (Kovats and Hajat, 2008), and as a result many national weather services have developed their own threshold definitions at a national and sub-national level (Kovats and Hajat, 2008). Furthermore, Hajat et al. (2010) showed that the threshold definitions used to issue heatwave

warnings are not always based solely on temperature but may include a range of meteorological parameters (e.g. humidity) depending on the method used.

Hajat et al. (2010) defined four different methods which are used globally to derive threshold values for heatwave alerts (Table 2.2). The synoptic classification and temperature-mortality methods (Methods 1 & 2, Table 2.2) are the most widely used methods, and both rely on epidemiological assessment of location-specific excess mortality during previous heatwave events (Hajat et al., 2010). Additionally, these methods can account for cumulative effects of heat-stress from a number of days and nights of extreme temperatures, which can be a key factor in heat stress levels (Clarke, 1972). In contrast, the temperature and humidity indices and the environmental stress index methods (Methods 3 & 4, Table 2.2) use implicit relationships between temperature and health to define thresholds for heat stress at a point-in-time and cannot therefore account for cumulative heat-stress over a number of days and nights (Hajat et al., 2010).

Within England and Wales the United Kingdom Met Office operates a heat-health warning system (UKMO, 2011) using the temperature-mortality method (Method 2, Table 2.2) (Department of Health, 2011). A heatwave warning is issued if it is predicted that temperatures will reach a given threshold for two consecutive days and the intervening night. Thresholds are spatially variable around England, with cooler northern regions having lower thresholds (e.g. North East; day maximum: 28°C, night minimum: 15°C), and warmer southern regions including London having higher thresholds (e.g. London: day maximum: 32°C, night minimum 18°C).

	Method 1: Synoptic classification	**Method 2: Epidemiological analysis of retrospective temperature and mortality data**	**Method 3: Temperature and humidity indices (e.g. Humidex)**	**Method 4: Physiologic approach (e.g. Environmental Stress Index)**
Theoretical basis	Categorises a variety of weather parameters into air mass types. Health response within each air mass type is determined epidemiologically using historical mortality data to identify 'oppressive' air mass types.	Thresholds based on temperature-mortality relationship estimated epidemiologically. French system seeks to identify days with >50% excess mortality in big cities and optimises sensitivity and specificity of thresholds.	Temperature and humidity integrated into one number to reflect a perceived temperature.	Heat loads identified using physiologic principles from heat budget equations which take into account mechanisms of heat exchange of the human body with its thermal environment.
Examples where the system is currently in operation	Philadelphia, Rome, Shanghai, Toronto, many other cities across US	France, England, Montreal	Humidex used across Canada	Germany
Accounts for cumulative effects (i.e. multiple days/nights of exposure to extreme temperatures)	Yes	Yes	No	No

Table 2.2: Table showing methods for defining heat-health watch system threshold values (source: Hajat et al. (2010) [Appendix I]).

One of the disadvantages of current heat-health warning systems is that they do not account for intra-urban spatial variability in temperatures and possible exposure (Dousset et al., 2011). However, it has been shown that at the city and sub-city scales, understanding the drivers for local climate modifications such as the UHI and their interactions with extreme events at a range of spatial and temporal scales is of key importance for future city climate adaptation and mitigation strategies (Solecki et al., 2005; Gaffin et al., 2008). Such understanding is necessary to quantify spatial variability in exposure to the heat hazard for city level heatwave action plans, as well as provide mitigation and adaptation options to reduce heatwave affects across the urban environment (Solecki et al., 2005; Gaffin et al., 2008).

2.3 The Urban Heat Island

The urban heat island (UHI) is an inadvertent climate modification which causes urban areas to be warmer than surrounding rural hinterlands (Figure 2.3) (Oke, 1973). The intensity of the heat island is spatially variable across the city, with the city core typically being 1-8°C warmer than the urban-rural fringe (Figure 2.3) (Oke, 1973). The UHI exists in the urban canopy layer (UCL) which is an atmospheric boundary layer between street and roof-top level (Figure 2.4) (Oke, 1987). The UCL is controlled by micro-scale climatic processes operating in the urban surface (e.g. thermal energy radiation from buildings). The UCL interacts with the urban boundary layer (UBL) which exists at the meso-scale, one to two kilometres above roof-height (Menut et al., 1999). Thermal radiation and warm air rising from the UCL causes warming in the UBL (Oke, 1987). Figure 2.4 shows a schematic where the UCL has warmed the UBL and regional airflow is causing non-isotropic spatial distribution of the warmer air in the UBL creating a heat plume downwind of the city.

The urban heat island is the result of changes in land cover and land use (including anthropogenic heat outputs) found in cities, compared to rural en-

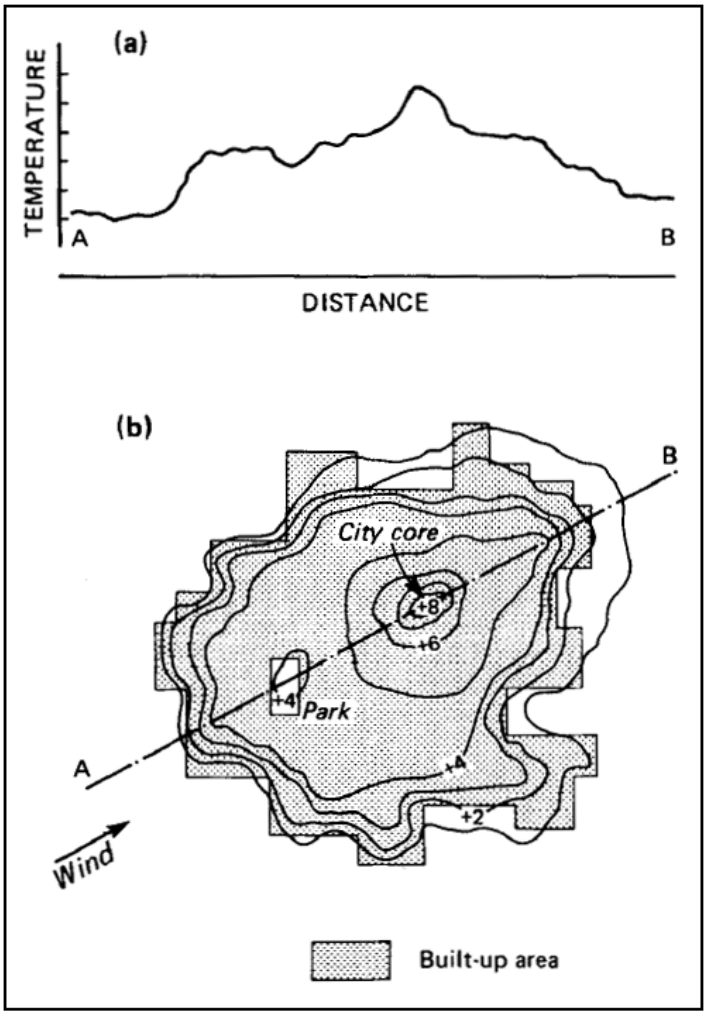


Figure 2.3: Urban Heat Island schematic; hypothetical representation of canopy layer temperatures across a mid-latitude city during calm and clear weather conditions (source: Oke (1982)).

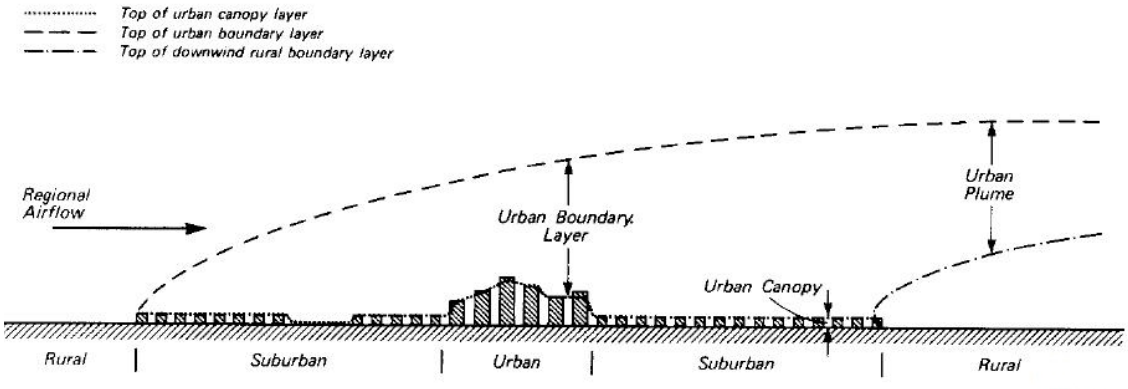


Figure 2.4: Schematic of atmosphere profile showing urban boundary layers and the urban heat plume effect (source: Oke (1976)).

vironments, which alter the energy balance between the land and the atmosphere (Oke, 1987). Table 2.3 lists the suggested causes of the urban heat island as proposed by Oke (1982, 1987). Oke (1982) cited examples from the literature (e.g. Unwin (1980)) which showed that for temperate latitude cities the heat island is greatest in the summer months, and lowest in the winter. For example, Unwin (1980) found that the average temperature difference between an urban weather station (Edgbaston) and a rural station (Elmdon) in Birmingham (UK) was 1.08°C (σ=1.64) during the summer (18th June to 9th September) but only 0.73°C (σ=1.40) during the winter (20th November to 19th January) between 1965 and 1974. From these results Oke (1982) deduced that it was unlikely therefore that anthropogenic heat output was the primary cause of urban warming, as this is normally greater in the winter than summer from heating of buildings. It is now accepted that the urban heat island is primarily caused by urban materials which capture and store high levels of solar energy (as compared to natural materials) and re-radiate this as thermal energy (Oke, 1987; GLA, 2006; Kolokotroni and Giridharan, 2008; Memon et al., 2009), (Table 2.3).

Altered energy balance terms causing urban heating	**Causes of energy balance alterations**
Increased absorption of short-wave radiation	Canyon geometry - increased surface area and multiple reflection
Increased long-wave radiation from the sky	Air pollution - greater absorption and re-emission of thermal radiation
Decreased long-wave radiation loss	Canyon geometry - reduction of sky view factor
Anthropogenic heat source	Building and traffic heat losses
Increased sensible heat storage	Construction materials - increased thermal admittance
Decreased evapotranspiration	Construction materials - increased waterproofing
Decreased total turbulent heat transport	Canyon geometry - reduction of wind speed

Table 2.3: Causes of the urban canopy layer urban heat island (source: Oke (1987), from Oke (1982)).

Increased absorption and thermal re-radiation of solar energy occurs because impervious surface materials such as concrete and asphalt have different radiative surface and thermal attributes to natural materials, such as a lower albedo and high heat capacity, which promotes solar energy absorption and

the capture and storage of thermal energy. For example the albedo of grass is between 0.16 and 0.26 (Budikova et al., 2010), whereas asphalt is typically lower, between 0.05 and 0.015 (ACPA, 2005). The thermal capacity (the amount of heat required to produce a unit change in temperature) of asphalt (0.92 kJ/kg K) is also often lower than natural materials (e.g. soil; 0.80-1.48 kJ/kg K, wood; 2.0-2.9 kJ/kg K) (Toolbox, 2012). As a result of these attributes urban materials convert a large proportion of incoming radiant energy into sensible heat (Oke, 1982). Furthermore, the geometry of urban infrastructure is a key driver of urban temperatures (Clarke, 1972; Oke, 1982). For example, the commonly found urban canyon, consisting of a roadway between rows of buildings can increase solar absorption by approximately 20% compared to a flat surface of the same material (Aida, 1982). The level of absorption increase depends on canyon depth, building density and solar zenith angle range (Aida, 1982). Additionally, the urban canyon acts as a radiator for thermal energy, trapping heat between the buildings instead of releasing it into the atmosphere, further warming the UCL (Oke, 1982).

A number of studies have shown that the variations in urban land-cover such as building geometry and density cause spatial variation in the urban heat island (Oke, 1981; Eliasson and Svensson, 2003; Di Sabatino et al., 2009; Hedquist et al., 2009). Harlan et al. (2006) showed that open space (i.e. where land cover was non-urban) was the strongest physical driver of urban temperatures across eight different neighbourhoods in the city of Phoenix, United States. The neighbourhoods were chosen to represent variations in socio-economic and physical variables, including median income, ethnic composition, age of housing stock, type of landscaping, population settlement density, neighbourhood openspace and vegetation density. Harlan et al. (2006) measured temperatures at 17:00 during the summer of 2003, including the five day heatwave between 12th and 16th July of that year. Neighbourhoods with a greater proportion of open-space (12-18%) exhibited cooler average temperatures during both summer and heatwave days (summer: 37.3 - 39.3°C, heatwave days: 40.2 - 44.4°C) than neighbourhoods with no open space (summer: 40.6 - 41.3°C, heatwave days: 40.9 - 45.6°C) (Harlan et al., 2006). It is strik-

ing that the average heatwave temperatures of the neighbourhoods with open space are equivalent to the average summer temperatures of the neighbourhoods with no open space. However, Harlan et al. (2006) did not quantify the strength of relationship between temperature and physical properties. Instead they used an outdoor human thermal comfort index (HTCI) metric to quantify the impact of socio-economic and physical variables on exposure of different neighbourhood populations to extreme temperatures (Harlan et al., 2006), and these results are further discussed in Sections 2.3.1 and 2.3.2.

Baker et al. (2002) undertook research in the same city (Phoenix, Arizona, USA) to investigate the effect of urbanisation on temperature. The city of Phoenix expanded rapidly in the 20th century, with a five-fold increase in urban land cover during this period (Baker et al., 2002). Using a long term temperature record from a weather station at the Phoenix Sky Harbour Airport between 1948 and 2002, Baker et al. (2002) showed that daily minimum temperatures had increased by 5.1°C and daily average increased by 3.1°C over this time period. The study notes that the average maximum temperature did not increase significantly during this period. Baker et al. (2002) state that the discrepancy between increasing minimum temperatures (found at night) and stable maximum temperatures is evidence of the increasing urban land cover re-radiating thermal energy after sunset, as is commonly seen with the urban heat island (Oke, 1987). Due to a lack of additional weather station records during this period Baker et al. (2002) did not attempt to directly quantify the relationship between the area of urbanisation and temperature increases. However, using a rural weather station (situated in arid, desert conditions, outside the city) from 1997-2000 the study showed that on average daily minimum temperatures at the airport were 6.1°C warmer than rural temperatures. Additionally, there was no difference in average daily maximum temperatures between the two stations (Baker et al., 2002), further supporting the authors' argument that urbanisation is causing the observed increase in minimum temperatures from nocturnal re-radiation of thermal energy absorbed by urban infrastructure.

However, the influence of urbanisation on urban temperatures is not simply

constant across the city (Eliasson and Svensson, 2003). For example, it has been observed that cool areas formed due to latent heat loss via evapotranspiration from vegetation as well as evaporative-latent heat loss and conduction from large water bodies, exist in some cities (Eliasson and Svensson, 2003; Kolokotroni and Giridharan, 2008). Solecki et al. (2005) state that such findings could be used in urban planning to encourage urban cooling via natural processes.

The spatial variability in temperature caused by changes in land cover across cities means that the urban heat island and urban temperature dynamics need to be quantified in a spatially explicit manner in order to capture the true variation of urban temperatures (Voogt and Oke, 2003; Wilby, 2003; Kolokotroni and Giridharan, 2008; Gaffin et al., 2008). To this end, Eliasson and Svensson (2003) used data from 30 weather stations in the city of Göteborg (Sweden) over an 18-month period to conduct a regression analysis between air temperatures and land cover classes. The use of a large number of stations spread over the city was designed to capture the inter-urban spatial variability (i.e. at the sub-city, neighbourhood scale) of temperature and the relationship with land cover class (Eliasson and Svensson, 2003). The study found significant temperature difference of up to 6.8°C existed between different city neighbourhoods (e.g. urban dense, industry, parks and greenspace).

Eliasson and Svensson (2003) classified measurements by weather conditions at time of measurement into two groups; clear and calm conditions ($\leq$2 octas and $\leq$3.3 m s^{-1}) and cloudy and windy conditions (>6 octas and > 3.3 m s^{-1}). During clear and calm conditions, Eliasson and Svensson (2003) found that distance from the sea and impervious surface cover had the highest correlations with temperature, with maximum regression coefficients (r^2) of 0.45 and 0.21 respectively. These findings are similar to the aforementioned study by Harlan et al. (2006) where a decrease in open space (natural land cover) in the city of Phoenix, appeared to increase neighbourhood temperatures (though this was not quantified directly), suggesting that there is a relationship between the area of urban material (impervious surface cover)

and near-surface temperature, observable in both desert and Scandinavian climates. Furthermore, during cloudy and windy conditions, Eliasson and Svensson (2003) found that elevation was the key factor in determining neighbourhood temperature (maximum r^2 0.31), suggesting that localised meteorological conditions are also a key driver of intra-urban temperature dynamics (Eliasson and Svensson, 2003).

2.3.1 Analysis of the Urban Heat Island

The urban heat island is commonly quantified using the urban heat island intensity metric (UHII) derived from terrestrial air temperature observations (Oke, 1973, 1987; Lee, 1992; Kim and Baik, 2002; Wilby, 2003; Di Sabatino et al., 2009; Wilby et al., 2011), although air temperature, the Human Thermal Comfort Index (HTCI) and the Temperature-Humidity Index have also been used (Baker et al., 2002; Eliasson and Svensson, 2003; Harlan et al., 2006). Table 2.4 shows a list of studies investigating the urban heat island, and the metrics used. Terrestrial air temperatures are normally measured between one and two metres above ground height, often called the 'near surface' or 'screen-level' temperature. For example, in the UK the standard screen-level height is 1.25 metres. The UHII is most commonly defined as the maximum difference between urban and background rural temperatures for a given point in time during one diurnal cycle (Equation 2.1) (Oke, 1987; Kim and Baik, 2002; Kolokotroni and Giridharan, 2008). However it should be noted that there is no clear consensus in the literature and UHII is more commonly generated using daily minimum and maximum temperature observations (Lee, 1992; Wilby, 2003; Gaffin et al., 2008; Jones and Lister, 2009; Wilby et al., 2011) (Table 2.4).

$$UHII = \Delta T_{u-r} \tag{2.1}$$

It is widely accepted that clear and calm conditions are favourable for urban heat island formation (Oke, 1973, 1987; Lee, 1992), as this ensures maximum

Study	Region/City	Data	Spatial distribution	Duration	Metric(s)
Balling and Brazel (1987)	Phoenix, United States	Weather stations	12 neighbourhoods	1949-1985	Temperature
Baker et al. (2002)	Phoenix, United States	Weather stations	Single urban station / 7 neighbourhoods for 1997-2000	1949-2002	Temperature THI
Eliasson and Svensson (2003)	Göteborg, Sweden	Weather stations	30 neighbourhoods	1998-2000	Temperature
Gaffin et al. (2008)	New York, United States	Weather stations	Single urban station	1900-2002	UHII*
Harlan et al. (2006)	Phoenix, United States	Temperature loggers	8 neighbourhoods	2003 (12 months)	Temperature THCI
Jones and Lister (2009)	London, United Kingdom	Weather stations	3 urban stations	1880-2005	Temperature UHII*
Kolokotroni and Giridharan (2008)	London, United Kingdom	Temperature loggers	80 urban stations, covering multiple neighbourhoods	2000 (12 months)	Temperature UHII
Lee (1992)	London, United Kingdom	Weather Stations	Single urban station	1962-1989	UHII*
Oke (1973)	North America (10 settlements)	Car transect (air temperature)	Multiple neighbourhoods	11 evenings, May 1970 to June 1971	UHII
Wilby (2003)	London, United Kingdom	Weather stations	4 urban stations	1959-1998	UHII*

Table 2.4: Table showing the metric used by studies investigating the urban heat island. *Note: *indicates UHII derived using daily minimum and maximum temperatures.*

solar energy input, and minimises turbulent mixing between boundary layers (Oke, 1987). Diurnally, the maximum UHII is often derived from air temperatures observed at night when heat is released by the urban fabric and reduced turbulent mixing allows warmer air to remain near the urban surface, while correspondingly rural temperatures are at their lowest; the diurnal maximum UHII under such conditions is often found between 22:00 and 06:00 (Figure 2.5). (Oke, 1973, 1987; Lee, 1992).

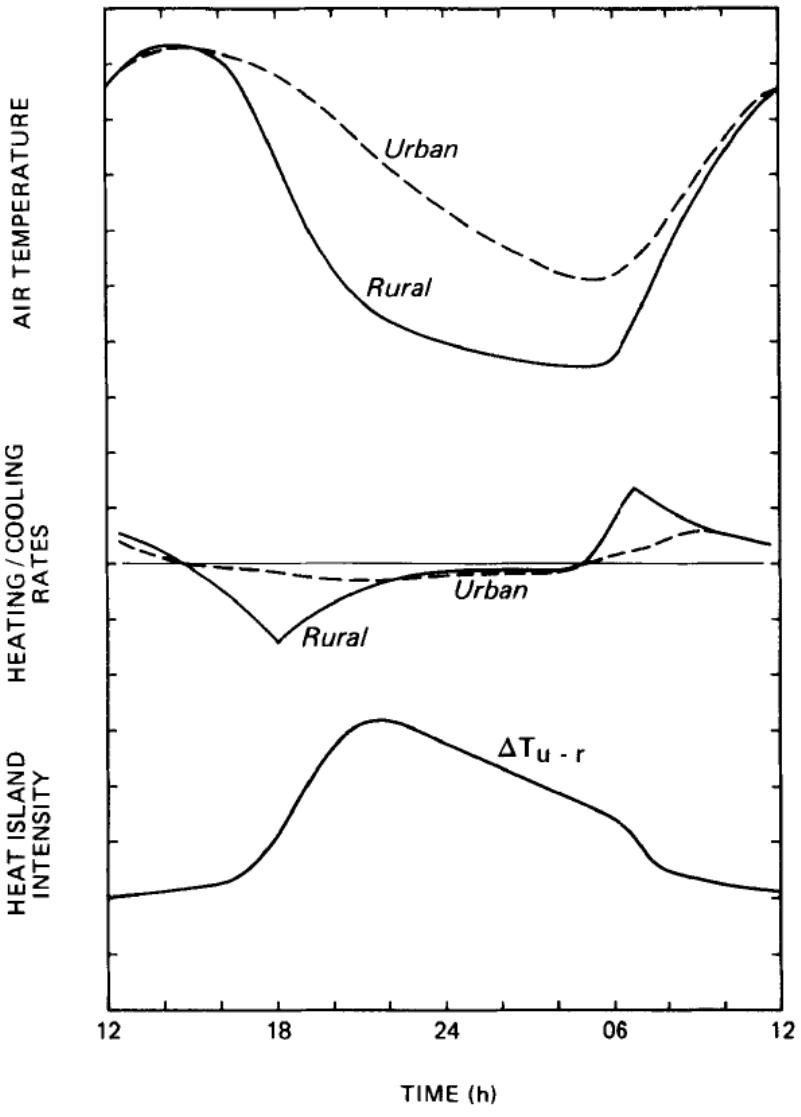


Figure 2.5: Typical temporal variation of urban and rural; air temperature, cooling/warming rates and urban heat island intensity under clear and calm weather conditions (source: Oke (1982)).

However, whilst it is well accepted that maximum air UHII can occur during

the night, the study by Kolokotroni and Giridharan (2008) contradicted this and found maximum air UHII to occur during daytime hours. Kolokotroni and Giridharan (2008) generated UHII using hourly air temperature observations from a bespoke network of 80 sensors across London over a 5 month period between May to September 2000. Hourly air temperature from a rural weather station outside of London was used as the reference station. The study found a single maximum daytime UHII of 8.9°C which occurred during partially cloudy periods but that average daytime UHII during partially cloudy conditions was 1.6°C. Interestingly the single maximum night-time UHII was 8.6°C which occurred during clear sky conditions while the average night-time UHII during clear sky conditions was 3.1°C. This suggests that whilst on average UHII may be greatest during the night, the overall maximum can occur during the day. It should be noted, however, that no standard deviation values were presented so the robustness of these results cannot be quantified.

Lee (1992) used two weather stations (one urban, one rural) to generate UHII time series for London over 28 years. Using observations from these stations to generate monthly average UHII between 1962 and 1990 Lee (1992) showed that over all four seasons of the year the UHII based on recorded daytime temperature were lower than those recorded at night-time (absolute values not presented in text). Furthermore, Lee (1992) found that for all seasons, daytime UHII was decreasing over the time-series (e.g. summer -0.007°C/year), whilst night-time UHII was increasing (e.g. summer +0.019°C/year). Lee (1992) hypothesises that this is the result of relative daytime 'cooling' in the urban environments compared to rural areas due to solar absorption by atmospheric pollution reducing daytime urban warming.

Wilby (2003), using data from the same two weather stations between 1959 and 1998, in contrast to Lee (1992) generated daily UHII. Wilby (2003) found that UHII based on maximum daytime temperatures increased during summer months (+0.035°C/year) and decreased or did not change during the other seasons. This suggests that the use of monthly averages of air temperature to generate UHII causes excess data smoothing, meaning that day/night differ-

ences in UHII are not properly captured.

Potentially, the differences between the studies by Lee (1992), Wilby (2003) and Kolokotroni and Giridharan (2008) indicate that generation of UHII using sub-diurnal hourly data as opposed to daily maximum, minimum, average or monthly average can influence the time of maximum UHII. In this regard it should be noted that there is no consensus in the literature of optimal temporal resolution for UHII metric generation (Gaffin et al., 2008). This is a recognised research gap (Kovats and Hajat, 2008), particularly with regard to the relationships between outdoor temperatures, the urban heat island and heat related mortality which are important to understand in order to quantify when the greatest population exposure to heat hazard occurs (Koppe et al., 2004).

A number of studies have implied that the utility of the UHII may also be somewhat limited in capturing the spatial variability of the urban heat island (Dousset, 1989; Dousset and Gourmelon, 2003; Jin et al., 2005; Rosenzweig et al., 2005; Cheval and Dumitrescu, 2009; Cheval et al., 2009; To et al., 2011). This is because in many cities weather station networks lack a sufficient spatial distribution to characterise intra-urban temperature dynamics (Eliasson and Svensson, 2003; Knight et al., 2010). Harlan et al. (2006) dispute the utility of studies like those of Lee (1992) and Wilby (2003), stating that the use of single or limited number of urban weather stations to quantify intra-urban temperatures could significantly under or overestimate the urban heat island in different parts of a city. However, it should be recognised that single point long time series studies have provided a useful insight into the long term trends of urban temperatures (Gaffin et al., 2008).

In contrast to spatially-sparse long-time series studies, Oke (1973) used temperature measurements from car transects to obtain spatially denser but temporally limited data. The transects covered 10 different North American settlements with population sizes from 1,000 to two million over 11 evenings throughout one year (May 1970 to June 1971) to investigate the relationship between settlement population size and UHII. In this case the UHII was cal-

culated using the average of temperatures recorded for rural and urban transects respectively. The results showed a positive relationship between settlement population and UHII with an r^2 value of 0.97 (Figure 2.6), indicating that UHII is directly proportional to settlement population (Oke, 1973).

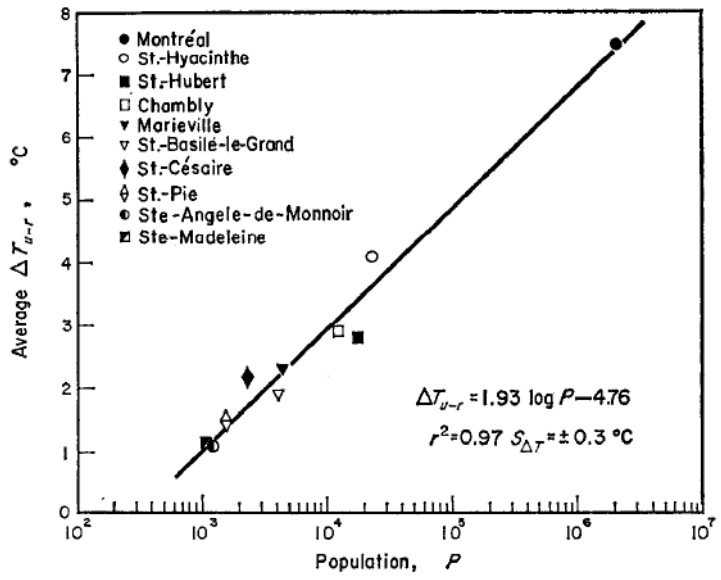


Figure 2.6: Relationship between UHII and population for 10 settlements on the St Lawrence Lowland during clear sky conditions (source: Oke (1973)).

Furthermore, Oke (1973) used UHII values from other studies in the literature, along with corresponding settlement population size to demonstrate that the same relationship could also be found with data for other North American (0.96 r^2) and European (0.74 r^2) settlements. Oke (1973) attributed the differences in the strength of the relationship between North American and European settlements to the greater spatial heterogeneity found in European cities as compared to the more homogeneous North America cities (Oke, 1973; Lee, 1992).

However, Oke (1973) did not utilise the transect data to investigate the spatial variability of temperatures across each of the settlements to quantify the drivers of the urban heat island variability. To address the issue of how spatially variable the UHI is within cities the aforementioned study by Kolokotroni and Giridharan (2008) used its temperature measurements to investigate the relationship between UHII and urban surface properties at each measurement location. The variables quantified by the study were; aspect ratio, sur-

face albedo, plan density ratio (ratio of building footprint to total neighbourhood area), green density ratio (ratio of green area to total neighbourhood area), fabric density ratio (vertical surface area to total neighbourhood area) and thermal mass. They found that maximum daytime UHII occurred in semi-urban areas at the edge of the city regardless of weather conditions. This was despite the fact that semi-urban areas had the overall lowest average temperatures compared to other land-use types such as the city-centre (absolute values not provided in text). Table 2.5 shows the maximum daytime UHII values measured by Kolokotroni and Giridharan (2008) for semi-urban, urban and city-centre land-use types. The table shows that the semi-urban zone UHII was up to 1.3°C greater than the city-centre during clear sky periods. In contrast the maximum nocturnal UHII was found in the urban area (8.6°C) which also had the greatest average nocturnal temperatures (absolute values not given in text) as one would expect for mid-latitude cities under calm conditions (Oke, 1987).

In order to try and explain these findings Kolokotroni and Giridharan (2008) investigated the relationships between the surface features and UHII using regression analysis. During all conditions (clear, partially-cloudy and cloudy) the three variables with highest correlation values with daytime UHII were; surface albedo (normalised correlation coefficients -0.24 to -0.30), aspect ratio (-0.17 to -0.23) and green density ratio (-0.23 to -0.38). Nocturnal UHII showed greater levels of correlation with surface properties, with surface albedo (normalised correlation coefficients -0.43 to -0.47), fabric density ratio (0.27 to 0.41) and green density ratio (-0.14 to - 0.20) being the top three variables across all conditions. These results suggest that surface albedo and green density ratio are important in controlling daytime and nocturnal UHII, though the relationship with aspect ratio and fabric density is less clear. Kolokotroni and Giridharan (2008) suggest that these results potentially demonstrate the influence of building materials and geometry on urban temperature. For example they state that 10 percent increases in surface albedo (lower solar absorption) and aspect ratio (narrower canyons restrict direct solar absorption and increasing shading) reduces temperatures

in London by up to 5.3°C (Kolokotroni and Giridharan, 2008). However, this interpretation should be used with caution as the study does not develop a true multi-variate model to test this hypothesis.

Description	Core (city centre) UHII (°C)	Urban UHII (°C)	Semi-urban UHII (°C)
Clear sky	6.1	6.9	7.4
Partially cloudy	7.2	8.0	8.9
Cloudy	7.9	8.4	8.8

Table 2.5: Maximum daytime UHII (°C) in three geographical zones in London during three climate states (wind velocity below 5 m/s) (source: Kolokotroni and Giridharan (2008)). *Note: standard deviation values not available.*

Earlier work by Balling and Brazel (1987) combined two databases of temperature measurements from short-term and long-term weather stations in the city of Phoenix (Arizona, USA). The study combined data from a spatially dense distribution of 29 stations which operated from 1980 to 1985 with a long term temperature record (1949-1985) from 12 further stations. In this way the spatial and long-term temporal characteristics of Phoenix's UHI could be investigated. The study revealed that temperatures were greatest towards the centre of the city (39-40°C) than surrounding suburbs (38-39°C), although the highest temperatures were measured in the dessert to the south west of the city (42°C). Over the time series the study found that six of the 12 long-term stations exhibited statistically significant (confidence interval not provided in the text) changes in recorded minimum temperature, five of which were increases (0.005-0.1°C/year) (Balling and Brazel, 1987). The airport station exhibited the greatest increase in temperature (0.1°C per year between 1949 and 1985), but no indication of variability over the time period is presented in the paper. Balling and Brazel (1987) concluded that the rapid increase in recorded temperature was directly related to and caused by the rapid urbanisation of Phoenix in the 20th Century, though the level of urbanisation was not quantified in the study.

Jones and Lister (2009) used a long time-series of weather station observations from 1907 to 2006 (station data availability was variable during this

period) in London (UK) (three urban stations, two rural) and by subtracting the rural time series from the urban stations, the study found an increase in the city's urban heat island intensity that was in addition to long-term increasing trends between 1951-1980. The intensification was most pronounced at London Heathrow airport which had an average increase of 0.4°C over the 29 years up to 1980. Thereafter no further increases in intensification were found. In contrast to London Heathrow, St James's Park (urban green space) and London Weather Centre weather stations showed no increase in intensity from the start of their available records in 1907 and 1974 respectively (Jones and Lister, 2009). Both of these stations are situated in the heart of London and Jones and Lister (2009) show that the intensity of the urban heat island in the centre of London has not increased during the available records, indicating that the UHI in central London must have developed before the start of the 20th century (Jones and Lister, 2009).

Contrary to these findings, Gaffin et al. (2008) subtracted yearly averages of air temperatures from Central Park in New York (USA) between 1900 and 2006 against an average of 23 non-urban stations and identified significant increase in heat island intensity through the time-series. The study found a 1.5°C temperature increase in Central Park between 1900 and 2000 (Figure 2.7(a)), and comparing this to the rural stations it is apparent that approximately one third of the increasing temperature trend (0.5°C) was attributable to intensification of the heat island during this period (Figure 2.7(b)). Gaffin et al. (2008) attributed the remaining two thirds (1°C) of increase to regional and global climate change.

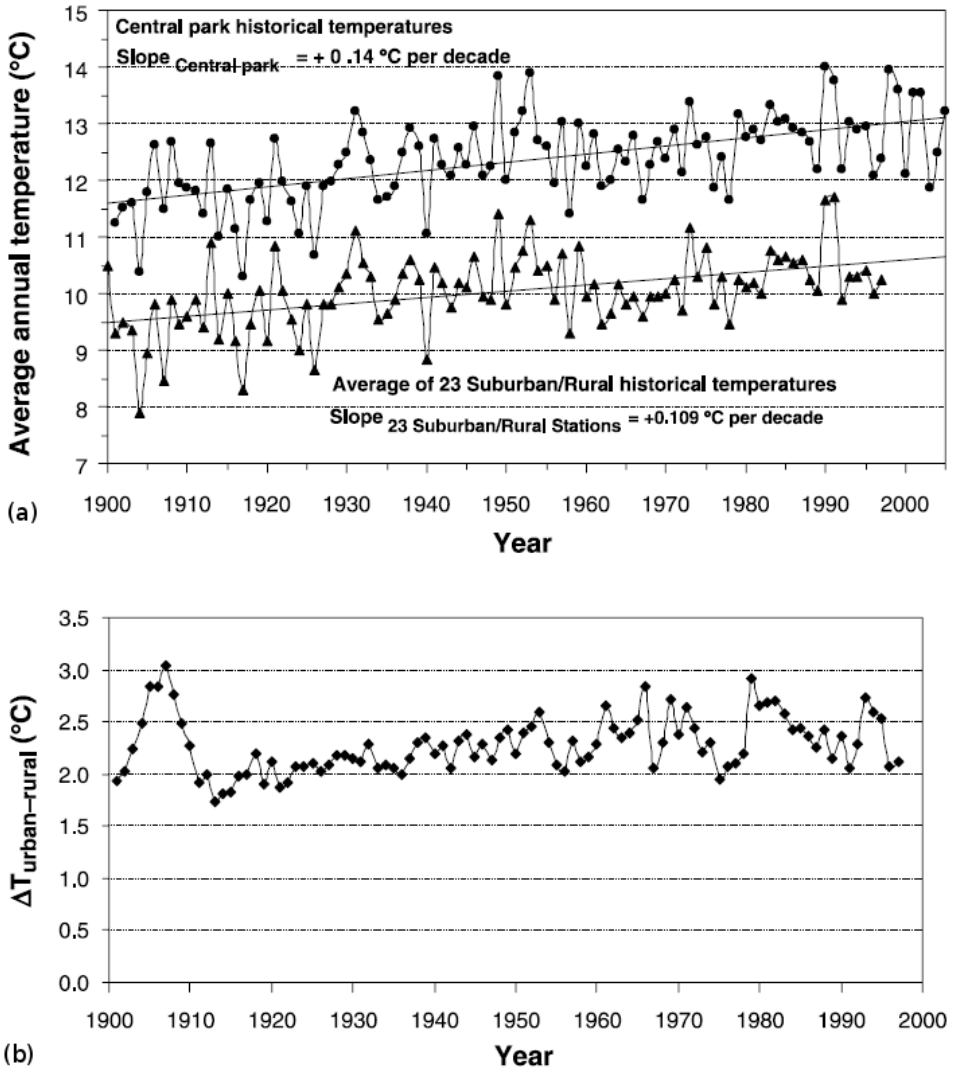


Figure 2.7: New York's urban heat island, UHII calculated from the difference between historical records for Central Park and an average of 23 suburban/rural stations (source: Gaffin et al. (2008)).

The differences between the studies by Gaffin et al. (2008) and Jones and Lister (2009) are striking, given that both used temperature observations from urban green spaces over a similar time period. A possible explanation of these disparate findings are that for New York a high degree of correlation between increasing temperatures in Central Park and a decrease in wind speeds measured in the city was found. Decreasing wind speeds in turn were thought to be the result of the developments of large buildings surrounding the park which increase urban canyon height and result in less turbulent heat exchange (Gaffin et al., 2008). In the case of London, however, less significant development has taken place since 1901 around St James's Park and London

Weather Centre, and hence had less influence on the UHII.

If the process of urbanisation and development is indeed responsible for the changes seen in Central Park it reveals not only the key role urban development dynamics may play in the UHI but also the importance of understanding the spatial aspects of this and the land cover, land use and spatial interactions that exist in the urban environment.

2.3.2 Impacts of the Urban Heat Island

The urban heat island effect compounds heatwaves in cities (Wilby, 2003; Solecki et al., 2005) and exposes urban populations to greater levels of heat hazard (Wilby, 2003; Dousset et al., 2011). The highest morbidity and mortality rates during extreme temperatures are found in urban areas (Harlan et al., 2006). Increased heatwave intensity as a result of global and local climate change such as the urban heat island is one of the immediate problems facing cities (Rosenzweig, 2011). Furthermore, it is accepted that the urban heat island also fosters development of additional biophysical hazards such as increased air pollution levels during heatwave events (Koppe et al., 2004; Solecki et al., 2005; Wilby, 2008). As a result, increased morbidity and mortality rates are often seen in urban hospitals during heatwaves, particularly in vulnerable populations such as the elderly or infirm (Fish et al., 1985; Kovats et al., 2006; Kovats and Ebi, 2006). For example Fish et al. (1985) reported a 61.8% increase in mortality (68 admissions) in elderly patients at Wandsworth Hospital (south-west London) during the 1983 heatwave (4th July-28th August) compared to the same period during the previous year. Patient deaths also increased at the hospital by 81.8% (18 deaths) during the heatwave (Fish et al., 1985),

Table 2.6, adapted from Kovats and Hajat (2008), demonstrates how recent heatwave events in England and Wales have severely affected the Greater London area with increased levels of mortality. During the 1976 heatwave there was a 15.4% increase in mortality in London, 5.7% greater than increases in the rest of the country. The heatwave of 1995 was even more severe,

with 184 (23% increase) excess deaths in London (12-21st July), which was double the percentage increase for England and Wales (11.2%, 768 deaths).

Heatwave event	**Increase in mortality in England and Wales**	**Increase in mortality in Greater London**	**Baseline measure**
1976	9.7%*	15.4%*	31-daily moving average of mortality in same year
1995 (12-21st July)	11.2% (768)	23% (184)	31-daily moving average of mortality in same year
2003 (4-13thAugust)	17% (2091)	42% (616)	Average of deaths for same period in years 1998 to 2002

Table 2.6: Heatwave events and their attributed mortality in England and Wales (adapted from: Kovats and Hajat (2008)). *Note: indicates * absolute number not available.*

During the 2003 August heatwave in Northern Europe, of the 2,091 excess deaths in England the largest short-term increase in mortality occurred in London with 616 excess deaths in a seven day period (a 43% increase from normal) (Johnson et al., 2005). Johnson et al. (2005) identified that exposure to urban temperatures was likely to be greater in London because of the urban heat island effect. However, Johnson et al. (2005) also showed that high concentrations of ozone and particulate matter with a diameter < 10 μm (PM_{10}), such as those recorded in London (excess ozone: 86 ug m^{-3}, excess PM_{10}: 26 ug m^{-3}) and the south East (excess ozone: 90 ug m^{-3}, excess PM_{10}: 20 ug m^{-3}) during the heatwave (4th and 13th August 2003) have been linked to increased cases of respiratory diseases. They estimated that 21-38 percent of the excess deaths in the 2003 heatwave could be attributed to the high ozone and particulate matter levels during this period.

Furthermore, Johnson et al. (2005) note that whilst the 2003 heatwave had

a severe impact on mortality in the UK, this was less than in France. During the most intense periods of the heatwave in August, lower temperatures and fewer days and nights of extreme temperatures were recorded in the UK compared to mainland Europe. For example, during a nine day period in the same month, there were 4,867 excess deaths in the Paris metropolitan area (Dousset et al., 2011), during which time minimum temperatures did not drop below 20°C and increased to 25.7°C (11th August) (Dousset et al., 2011). In contrast, the highest minimum temperature ever recorded in the UK is 23.9°C (4th August 1990, Brighton) (Burt, 2004b), and during the August 2003 heatwave greatest minimum in London was 20.8°C recorded on 6th August (Burt, 2004b).

The differences in mortality and night-time temperature between London and Paris during the August 2003 heatwave correspond with a number of studies which have identified high night-time urban temperatures as causing the greatest thermal stress in humans (e.g. Solecki et al., 2005). Kovats and Hajat (2008) note that heatwave mortality is more sensitive in urban than rural and suburban areas and suggest that this is because the heat island magnifies night-time temperatures and reduces the ability for nocturnal cooling, which is a well known cause of heat stress (Clarke, 1972). As a result of the apparent relationship between night-time urban temperatures and increased mortality, a number of studies have focused on night-time temperatures, especially because the urban heat island intensity is traditionally greatest during the night in mid-latitude cities (Oke, 1987; Kovats and Hajat, 2008).

However, the relationship between increased mortality and increased nocturnal temperatures is not a consistent finding within the literature (Baker et al., 2002; Kolokotroni and Giridharan, 2008). For example Baker et al. (2002) examined the impacts of increased UHI as a result of urbanisation for Phoenix (Arizona, United States) and demonstrated that daytime temperatures were equally important, as the greatest number of heat related deaths in Arizona occur during the daytime (approximately 30 a year, 13 times the national average) (Baker et al., 2002). Baker et al. (2002) hypothesised that this was due to outdoor exertion during the extremely hot summer days (defined as

$\geq$38°C for their study) experienced in an arid city such as Phoenix. Using the Temperature-Humidity Index (THI) (see Baker et al. (2002)) to analyse changes in heat stress, Baker et al. (2002) showed that the number of misery hours per day (hours when the temperature is above a predefined comfortable level) had increased as a direct result of urbanisation between 1948 and 2000 and therefore this increased the number of heat related deaths occurring over summer periods.

The aforementioned study by Harlan et al. (2006) used the Human Thermal Comfort Index (HTCI) metric (see Harlan et al. (2006)) to examine neighbourhood level response to a heatwave event in Phoenix (USA). The study found that the neighbourhood temperature differences increased during extreme temperatures events, and that high building-density and sparsely vegetated neighbourhoods correlated with the location of highest temperatures and HTCI values. Open space (percentage) was the physical variable with the highest correlation to HTCI (-0.65 $p<0.5$). However, social factors had higher correlations; for example percentage of houses with swimming pools had the highest correlation with summer HTCI (-0.83 $p<0.1$), suggesting that socio-economic neighbourhood characteristics are the greatest indicator of locations with high HTCI (Harlan et al., 2006).

Harlan et al. (2006) also found that the distribution of temperatures (as quantified by HTCI) were not linear with proximity to city-centre as is traditionally perceived to be the case (Oke, 1973, 1987; Kolokotroni and Giridharan, 2008) but were highly correlated with surface properties at observation location, especially vegetation density and the amount of open space (see above). The findings of Harlan et al. (2006) are similar to those of Eliasson and Svensson (2003). However, in contrast, Harlan et al. (2006) found the spatial variability of temperatures at the neighbourhood scale was independent of overall location in the city. This suggests that temperatures (and subsequent derived heat stress indicators) observed by Harlan et al. (2006) in the arid city of Phoenix are not controlled by city topography and surrounding geography as they were in the coastal city of Göteborg investigated by Eliasson and Svensson (2003). In this respect Harlan et al. (2006) demonstrates the importance

of quantifying temperatures and heat stress spatially across cities to identify areas where heat hazard is greatest. Harlan et al. (2006) states that if such areas can be recognised then they can be targeted for mitigation and adaptation options, such as in the case of Phoenix, reflective roofing and paving, lower anthropogenic heat emissions and provision of shade and pool access to all neighbourhoods.

2.4 Thermal remote sensing of cities

2.4.1 Introduction

Quantifying and monitoring the drivers and impacts of rapid urbanisation is often restricted by spatially limited terrestrial observations (Cheval and Dumitrescu, 2009; Miller and Small, 2003). However, increasingly, remote sensing and in particular, satellite Earth observation are providing an alternative data source which can be used to facilitate new tools and approaches for understanding the urban environment (Miller and Small, 2003). Traditionally, aerial photogrammetry has provided data for cartographic mapping and planning in cities (Donnay, 1999). However, since the 1980's a number of papers have cited the advantages of using satellite Earth observed data sets for urban monitoring, particularly with regards to the increased repeatability of observations, higher revisit times and lower costs of data acquisition (Forster, 1985; Owen et al., 1998; Donnay, 1999; Mesev, 2003).

Applications of such data include land cover/land use mapping (e.g. Brugioni (1983); Khorram et al. (1987); van der Linden and Hostert (2009)), land use change monitoring (e.g. Xian et al. (2008)), population estimation (e.g. Brugioni (1983); Lo (2003)) and urban surface temperature monitoring (e.g. Miller and Small (2003); Voogt and Oke (2003)). Miller and Small (2003) state that Earth observed data has a number of unique characteristics which support its use as an alternative or supplementary data source with which to quantify urban environmental variables, including; a broad spatial coverage, repeatable

observations and more recently an archive of data stretching back between 20 to 30 years which can be used to assess temporal change (Miller and Small, 2003).

Data from a number of different Earth observation sensors has been employed within urban studies for a range of purposes (Mesev, 2003). For example, a number of studies have investigated how Earth observed data may be employed to measure population dynamics. Brugioni (1983) hypothesised that it should be possible to conduct a census of the United States using Earth observed imagery and empirically derived relationships between building density, population density and population distribution. Furthermore, Brugioni (1983) proposed that a distance decay function could be used to model the reduction in population density with distance from a city-centre. With respect to Brugioni (1983), Lo and Quattrochi (2003) used an allometric growth model first proposed by Tobler (1969) to estimate population in Atlanta, United States, based on available area for development derived from Landsat Multispectral scanner (MSS) and Thematic Mapper (TM) scenes. However, Lo and Quattrochi (2003) showed large errors (18.12%) in population estimates compared to census data, and suggest that the method requires better land-use classification, generated using higher spatial resolution imagery.

Earth observation has been widely used to perform urban land cover classification (e.g. van der Linden and Hostert (2009); Xian et al. (2008)), traditionally achieved using imagery in the visible (0.5-0.7µm), near infra-red (0.74-1.4µm) and thermal wavelengths (8-15µm) (Owen et al., 1998; Dousset and Gourmelon, 2003; Lo, 2003; Latifovic et al., 2005; Xian et al., 2008). A number of studies have also investigated new sensor technologies such as hyperspectral and RADAR imagery to improve urban land use classification (e.g (Dong et al., 1997; Stabel and Fischer, 2001; van der Linden and Hostert, 2009)). For example, van der Linden and Hostert (2009) investigated the utility of the HyMap hyperspectral sensor (0.4-2.5µm, 4.5m on-nadir) to differentiate spectrally similar built-up and non-built up land cover classes, traditionally mis-classified with broad-band sensor imagery (van der Linden and Hostert, 2009). However, the study found large errors in an urban land use

classification of Berlin using the HyMap data, and highlighted that these errors were the result of urban vegetation and complex building geometry distorting the spectral signal of urban land cover (van der Linden and Hostert, 2009). The study recommended that hyperspectral images used for urban land cover classification should have a small field of view to minimise geometric distortion and potential miss-classification (van der Linden and Hostert, 2009). For cities in tropical regions where high levels of cloud cover prevent optical, infrared or thermal data from being captured Dong et al. (1997) and Stabel and Fischer (2001) have cited the advantages of using RADAR (Radio Detection and Ranging) imaging to delineate urban land-cover extents and measure urban growth between multiple images in a time-series. Stabel and Fischer (2001) examined the utility of repeat-pass synthetic aperture RADAR interferometry (InSAR) from the European Remote Sensing Satellites 1 and 2 (ERS-1 and ERS-2) to delineate urban and rural land-cover for Taipei (Republic of China) and Cairo (Egypt). The study found mixed results, the InSAR imagery could delineate the extents of Taipei, but could not define the urban-rural boundary for Cairo. Stabel and Fischer (2001) concluded that whilst RADAR imagery offers a potentially unique data source for urban applications, analysis and prepossessing methods are still under development, and require further improvement.

In contrast to hyperspectral and RADAR imaging, the use of thermal Earth observation to study the thermal characteristics of cities is widespread (Table 2.7). Thermal Earth observation has been used to map the urban heat island (Matson et al., 1978; Dousset, 1989; Roth et al., 1989; Pongracz et al., 2006), perform land cover classifications (Dousset and Gourmelon, 2003; Xian et al., 2008) and provide inputs for urban surface modelling (Kato and Yamaguchi, 2005; Xu et al., 2008). Primarily, thermal Earth observation has been used to characterise the spatial structure of urban temperatures which are often difficult or impossible to ascertain from terrestrial observations (Roth et al., 1989; Nichol, 1996; Tomlinson et al., 2012). Thermal Earth observation has also been used to relate surface temperature variability to urban surface characteristics (Voogt and Oke, 2003; Tran et al., 2006). As such, thermal Earth

observations are considered a valuable supplementary data source for terrestrial measurements to quantify urban 'hot spots' and 'cool islands' (Nichol, 1996), as well as identify areas of greatest exposure to the urban heat island and extreme temperature events (Gaffin et al., 2008). Increasingly, thermal Earth observation is also being used to provide data for climate mitigation strategies in urban areas (Pongracz et al., 2006), as surface temperature is a key controlling variable in the energy exchanges that take place within urban areas, modulating air temperatures within the urban canopy layer (Voogt and Oke, 2003).

Within the context of this study (see Section 1.3), a review of the literature shows that studies using thermal Earth observation in urban areas can be divided into four research groups: quantifying urban temperature dynamics (Section 2.4.3), modelling the relationship between urban surface and air temperatures (Section 2.4.4), quantifying the urban heat island (Section 2.4.5), and analysis of heatwave events using EST and UHII (Section 2.4.6). Table 2.7 provides an overview of the literature reviewed in each group (studies investigating heatwaves are noted in their respective temperature and UHII groups).

Table 2.7 shows the range of sensors employed for analysis of urban temperature dynamics, with broader spatial resolution sensors such as AVHRR (1100m on nadir) and MODIS (1000m on nadir) being the most widely used. Older sensors such as AVHRR and MODIS also provide a larger number of scenes for analysis and Table 2.7 shows that data from these sensors has been used in greater volumes across all four research groups. The newer ASTER sensor (90m spatial resolution on nadir) has also been widely used, though with fewer scenes. Section 2.4.2 discusses the trade-off between between spatial resolution and re-visit time/time-series availability. Table 2.7 also highlights the limited number of studies employing thermal Earth observation to undertake research directly related to urban heatwave analysis, as discussed in Section 2.4.6.

Table 2.7: Studies using Earth observed thermal imagery for analysis of urban temperature dynamic, grouped by research area.
*Note: * indicates study investigated a heatwave event.*

Study	**Region / City**	**Sensor**	**No. scenes**	**Resolution (metres)**	**Time period**
Quantifying urban temperature dynamics					
Dousset and Gourmelon (2003)	Los Angeles, United States & Paris, France	AVHRR	107	1100	July-August 1984-1985 and 1998
Dousset et al. (2011)*	Paris, France	AVHRR	61	1100	July-August 2003
Nichol (2005)	Hong Kong, China	ASTER/Landsat ETM+	1/1	90/60	September and October 2001
Nichol et al. (2009)	Hong Kong, China	ASTER	1	90	January 2007
Owen et al. (1998)	Pennsylvania, United States	AVHRR	8	1100	1985, 1998, 1991 and 1994
Roth et al. (1989)	Vancouver, Seattle & Los Angeles, United States	AVHRR	11	1000	1985
Stathopoulou and Cartalis (2009)	Athens, Greece	AVHRR/Landsat TM	1/4	1100/120	2004-2005

Modelling EST-air temperature relationships					
Dousset (1989)	Los Angeles, United States	AVHRR	84	1100	August 1984 and 1985
Czajkowski et al. (2000)	Oklahoma, United States	AVHRR	6	1100	August 1994
Fung et al. (2009)	Hong Kong, China	ASTER	1	90	January 2007
Nichol et al. (2009)	Hong Kong, China	ASTER	1	90	January 2007
Prihodko and Goward (1997)	Kansas, United States	AVHRR	31	1100	May-October 1987
Vogt et al. (1997)	Andalusia, Spain	AVHRR	148	1100	1992
Urban Heat Island Intensity analysis					
Cheval and Dumitrescu (2009)	Bucharest, Romania	MODIS	609	1000	July 2000-2006
Cheval et al. (2009)*	Bucharest, Romania	MODIS	765	1000	July 2000-2007
Tran et al. (2006)	Asia (eight cities)	MODIS/Landsat ETM+	4/2	1000/30	2001-2003
Tomlinson et al. (2012)*	Birmingham, United Kingdom	MODIS	63	1000	June-August 2003-2009
Matson et al. (1978)	North and mid United States	VHRR	1	1100	July 1977
Streutker (2002)	Houston, North America	AVHRR	48	1100	1998-2000
Streutker (2003)	Houston, North America	AVHRR	207	1100	1985-1987 and 1999-2001
Pongracz et al. (2006)	Budapest, Hungary	MODIS	-	1000	2001-2003

2.4.2 Thermal Earth observation concepts

Thermal Earth observation sensors work by measuring the emitted electromagnetic radiation in the thermal infra-red wavelengths between 8 to 12μm of the electromagnetic spectrum. Thermal radiation is emitted by all objects with a kinetic temperature greater than absolute zero and the distribution of wavelengths of emitted energy is directly proportional to the temperature of an object's surface layer (Planck's law) (Mather, 2004). Planck's law describes the distribution of the spectral radiant flux density for a given wavelength (i.e. emitted radiation per square metre ($Wm^{-2}sr^{-1}$)) as a function of temperature of a blackbody and allows the derivation of the blackbody temperature of a surface (Lillesand et al., 2004; Mather, 2004). Satellite, airborne and terrestrial thermal sensors measure emitted thermal radiation from the Earth's surface to estimate the surface temperature of objects within the sensor's field of view (Mather, 2004).

The thermal emittance and absorption of objects varies as a function of their surface radiant properties and thermal capacity (Lillesand et al., 2004; Mather, 2004). Emissivity is the measure of an object's efficiency to emit radiation at a given temperature compared to a blackbody at the same temperature (Lillesand et al., 2004). When using thermal remotely sensed data emissivity of surface features must be applied to blackbody temperatures to derive an estimate of surface temperature (EST) (Lillesand et al., 2004). Over large homogeneous surfaces such as oceans a constant emissivity can be applied to derive sea surface temperature (Gentemann et al., 2003). However, over heterogeneous land cover, such as urban areas, emissivity values representative of land cover must be applied if an absolute estimate of temperature is required (Caselles et al., 1997).

Due to operational constrains a number of studies have used relative blackbody temperatures for qualitative analysis (e.g. Roth et al., 1989; Streutker, 2002). Emissivity data can come from ancillary sources, remotely sensed optical based land classification or via an emissivty-temperature separation algorithm (Voogt and Oke, 2003). Uncertainty in emissivity can lead to EST

errors of up to 3°C (Dash et al., 2002). As such, the requirement for emissivity has restricted the utility of the thermal Earth observed data for land surface studies (Voogt and Oke, 2003), as this error is potentially greater than temperature variations between different surface types of interest.

The measured thermal radiance recorded by satellite and airborne thermal sensors can also be subject to atmospheric attenuation due to water vapour and atmospheric gasses (Dash et al., 2002). For example Goetz et al. (1995) found an error of 4.3°C between atmospherically corrected and uncorrected Landsat Thematic Mapper temperature measurements. Atmospheric corrections can be applied using a number of methods, with either ancillary or optical image data (Forster, 1985; Cooper and Asrar, 1989; Erbertseder et al., 1999; Hadjimitsis et al., 2004). Methodologies for emissivity and atmospheric correction of thermal Earth observed imagery are discussed in Chapter 3 (Data acquisition and pre-processing). For the purposes of this chapter, unless otherwise stated the studies presented deal with applications of estimated surface temperature (EST) derived from thermal Earth observation which have all been corrected for emissivity and atmospheric attenuation.

Since the 1980's a number of studies have demonstrated the utility of satellite Earth observed land surface temperatures as an alternative or supplementary data source for urban temperature analysis (Table 2.7). Over this period a range of sensors have been developed with thermal bands of varying spatial and spectral resolutions; from low spatial and spectral resolution sensors such as the AVHRR (1.1Km on nadir, two thermal bands) to newer higher spatial and spectral resolution thermal sensors such as ASTER (90m at nadir and with five thermal bands). However, due to technological constraints, satellite thermal sensors are unable to capture both high spatial and high spectral resolution data with a low re-visit time (Stathopoulou and Cartalis, 2009). As a result, satellites with higher spatial and spectral resolutions have a much longer revisit time (Table 2.8) (Stathopoulou and Cartalis, 2009). Re-visit time is also a function of the number of satellites carrying the same sensor. For example, the AVHRR sensor is currently deployed on six satellites meaning that it can capture up to four scenes per day for the same location of the

Earth (Cracknell, 1997) (Table 2.8). In contrast the Moderate Imaging Spectroradiometer (MODIS) is deployed on the Terra and Aqua satellites and has a revisit time of 1 to 2 days (Table 2.8). Therefore, studies wishing to analyse urban thermal dynamics have to balance the requirements for high-spatial and spectral resolutions against lower revisit time, and hence fewer scenes for analysis (Stathopoulou and Cartalis, 2009).

Sensor / Platform	**Spatial resolution (m)**	**Spectral resolution of thermal wavelengths (μm)**	**Maximum temporal resolution**
AVHRR / NOAA	1100	band 4: 10.3-11.3, band 5: 11.5-12.5	4 times daily
MODIS / Terra & Aqua	1000	band 31 to 36: 10.78-14.39	1-2 days
TM / Landsat 5	120	band 6: 10.4-12.5	16 days
ASTER / Terra	90	band 10 to 14: 8.125 - 11.65	16 days
ETM+ / Landsat 7	60	band 6: 10.4-12.5	16 days

Table 2.8: Technical characteristics of operational satellite thermal sensors showing spatial, spectral and temporal resolution. Adapted from Stathopoulou and Cartalis (2009).

As a result of the trade-off between high spatial resolution and temporal revisit time, Nichol (2009) and Stathopoulou and Cartalis (2009) have suggested the use of downscaling low-spatial resolution thermal Earth observation data to derive higher resolution thermal data, with a low revisit time. Downscaling is the process of converting low resolution data to higher spatial resolution using ancillary data (Stathopoulou and Cartalis, 2009). Typically a low resolution image is merged with a high resolution image generated from a different sensor, or from ancillary data, to form a new higher resolution image. For thermal data Nichol (2009) proposed an emissivity modulation method for spatial enhancement. The module works during pre-processing when a low resolution thermal scene is being used to generate estimated surface temperature (EST). A high resolution scene containing emissivity values is used to perform a sub-pixel correction for emissivity (from blackbody temperature to EST), creating a new thermal image with the same (higher) resolution as the emissivity correction scene (Nichol, 2009). In this manner the method cor-

rects for surface emissivity and enhances spatial resolution simultaneously (Nichol, 2009).

Nichol (2009) applied this technique to band 13 (10.25-10.95μm) from a single night-time ASTER scene (90m spatial resolution) of Hong Kong (China), using an emissivity correction map with 10m resolution, to derive an EST image with 10m resolution. A 90m resolution EST scene was also created for comparison. In relation to in-situ surface temperatures, Nichol (2009) showed that the EST scene at 10m spatial resolution had a much stronger relationship (r^2=0.71) than the 90m EST (r^2=0.58). Nichol (2009) suggests that these results show that an emissivity based downscaling model can improve the spatial resolution of thermal data, allowing it to capture intra-urban variations in temperature with greater accuracy. However, Nichol (2009) notes that the method is less likely to work for thermal scenes captured during the daytime, when surface factors such as albedo and shade are likely to have a greater influence on intra-urban surface temperatures than emissivity. The implications of such downscaling when employed for analysis of the EST-air temperature relationship (e.g. Nichol et al. (2009)) are discussed further in Section 2.4.4.

2.4.3 Quantifying urban temperature dynamics

Due to the aforementioned advantages of thermal remote sensing (see Section 2.4.1) compared to terrestrial measurements (spatial-completeness, homogeneous coverage and repeatability (Miller and Small, 2003)), thermal Earth observed data have been used by a number of studies to quantify intra-urban variations in temperature in a spatially complete manner, that has not been attainable using weather stations (Matson et al., 1978; Dousset, 1989; Roth et al., 1989). Roth et al. (1989) used eight daytime and three night-time georectified scenes from the Advanced Very High Resolution Radiometer (AVHRR) thermal band 4 (10.5-11.5μm) to examine the urban heat islands of three coastal cities of Western North America (Vancouver, Seattle and Los Angeles). The 11 scenes were captured between the 29th November 1983 and 14th

February 1986, during atmospherically stable, cloud-free anti-cyclonic conditions. Of the 11 scenes only two (both daytime) covered Seattle and Los Angeles, with the remaining nine scenes covering Vancouver, the primary focus of the study. Due to the requirements for information of atmosphere content (e.g. water vapour) and emissivity, Roth et al. (1989) did not perform atmospheric or emissivity corrections on the scenes and instead used the at-sensor brightness temperatures to quantitatively examine urban temperatures (Roth et al., 1989). Despite the limited number of scenes (potentially reducing the reliability of results), the study showed that in all scenes land use (derived from terrestrial mapping) appeared to correspond with urban temperatures. For example, the warmest locations in the images were found in areas of dense urban or industrial development, although no quantitative correlation analysis between land-use and intra-urban temperatures was conducted (Roth et al., 1989).

Later analysis by Dousset and Gourmelon (2003) agreed with the initial findings of Roth et al. (1989) and showed that the industrial areas of Los Angeles were up to 7°C warmer than surrounding rural hinterlands. These findings were based on ESTs derived from 85 AVHRR scenes spanning the months July to August from 1984 and 1985, grouped by overpass time, and temporally averaged over the time-series. Figure 2.8 (from Dousset and Gourmelon (2003)) shows the temporal averages of EST at selected locations for each of the local times of AVHRR overpass (the number of scenes used to create each average at the time of overpass is not presented in the text). Within the figure the diurnal cycle can clearly be seen, including temperatures in the city centre ("downtown") exhibiting a > 20°C increase in the afternoon (14:50) compared to the early morning (04:25) (Dousset and Gourmelon, 2003). Rural areas outside of the city ("Chino fields") exhibit a similar range of temperatures through the diurnal cycle and which are approximately 5°C cooler than urban areas (Figure 2.8).

Using a land cover classification derived from a SPOT-HRV image (Système Probatoire d'Observation de la Terre - high resolution visible) Dousset and Gourmelon (2003) showed that within Los Angeles large urban green areas

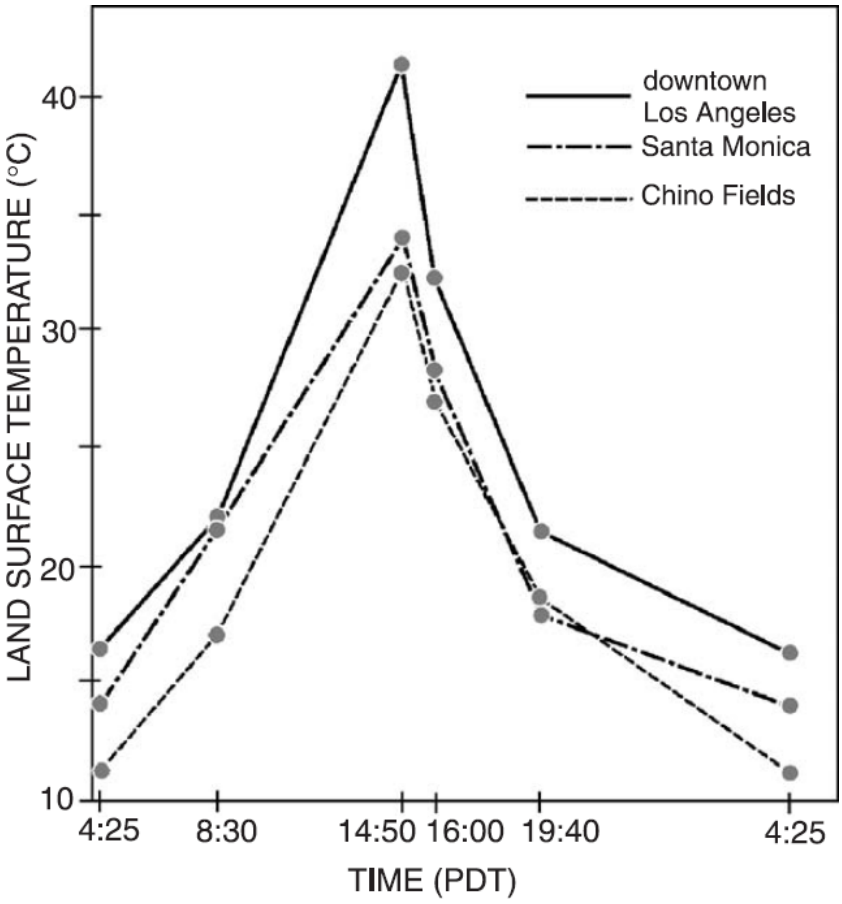


Figure 2.8: EST for Los Angeles and surrounding areas derived from a temporal averages of 84 AVHRR scenes captured between July-August, 1984-1985 (source: Dousset and Gourmelon (2003)).

(total surface area not provided in text) created cool-islands ranging between 2.2-5.0°C compared to surrounding urban temperatures. However, no direct relationship between green area size and temperatures were derived in the study for Los Angeles.

Dousset and Gourmelon (2003) used a further 22 AVHRR scenes, processed to EST to examine intra-urban temperatures in Paris, and in particular to quantify the relationships between intra-urban temperatures, building density and vegetation cover. Such information is important to understand how differences in surface properties can influence EST so that future climate/heatwave adaptation and mitigation strategies can account for the influence of urban surface properties. Dousset and Gourmelon (2003) derived six land cover classes for the Paris area using a SPOT HRV image with unsupervised classification; water, urban, densely built, sub-urban residential, light bare soil, and vegetation. Using the six classes a percentage built-up density within each AVHRR pixel (SPOT resolution is 20m) was then generated and used to investigate the correlation between urban built density and EST.

Figure 2.9 is an extract from Dousset and Gourmelon (2003) showing a correlation plot for building density and day/night ESTs derived from AVHRR scenes. No measure of correlation was derived, although based on the results in Figure 2.9 , Dousset and Gourmelon (2003) state that "The nighttime distribution of LST [land surface temperature] is well correlated with increasing density of buildings from the suburbs to downtown...". Furthermore, the study states that daytime temperatures are also correlated, but with higher variations due to large fluxes in temperatures during the daytime (as seen with the diurnal representation of temperatures in Los Angeles above). However, without a measure of correlation, based on the results shown in Figure 2.9, the interpretation of these results by Dousset and Gourmelon (2003) is doubtful, and further work is required to prove that a statistically significant relationship exists.

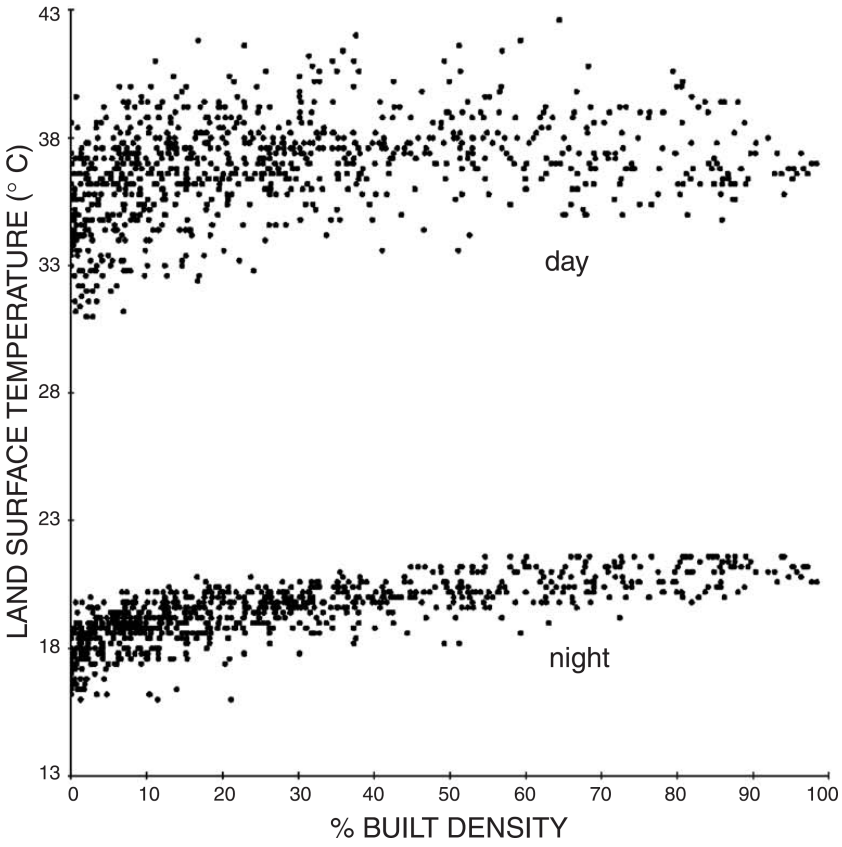


Figure 2.9: Joint distribution of percentage of built-up density for Paris and surrounding area with average night-time and daytime AVHRR EST values (source: Dousset and Gourmelon (2003)).

In an attempt to better understand the broad-scale relationships shown by Roth et al. (1989) and Dousset and Gourmelon (2003), thermal satellite sensors with higher spatial resolution (<500m) have been used to examine the micro-climatic effects of urban morphology (e.g. Nichol, 2005). Nichol (2005) used a daytime Landsat ETM+ scene (60m spatial resolution) and a night-time ASTER scene (90m spatial resolution), both processed to EST, to examine micro-climatic temperature variations across the Western New Territories Region of Hong Kong (Nichol, 2005). The daytime EST from the Landsat ETM+ scene showed variations of up to 8°C across the urban area. Using ancillary land use classification data including building and road outlines and an aerial image, the study showed that daytime ESTs appeared to be related to sky view factor (SVF). The study showed that areas with low SVF (SVF values not presented in text) had lower temperatures (~35°C) than areas with high SVF (~37-38°C) (Nichol, 2005). Furthermore, Nichol (2005) showed that locations shaded by buildings at the time of overpass were up to 4°C cooler

than adjacent non-shaded areas.

In contrast to the daytime EST values in Nichol (2005), the night-time ESTs from the ASTER scene overpass (21:40) showed almost complete uniformity across the urban area of the Western New Territories Region, with intra-urban temperature variations of less than 2°C. Nichol (2005) states that the marked difference in intra-urban temperature variation between day EST (Landsat) and night-time EST (ASTER) indicate that micro-climatic variations are more pronounced during the day. This suggests that urban morphologically derived variation in temperatures is driven primarily by exposure of the urban surface to solar input during the day. Further, Nichol (2005) hypothesis that an ASTER scene captured later in the night would have shown greater variations in temperature, as urban surfaces cool at different rates depending on their thermal capacity and exposure to solar infrared and thermal radiation during the day.

Nichol (2005) also examined the influence of the sea on cooling of the urban areas, which has previously been found to be a key driver of intra-urban temperatures (Roth et al., 1989; Dousset, 1989; Dousset and Gourmelon, 2003; Eliasson and Svensson, 2003), in terms of air temperatures (Eliasson and Svensson, 2003) and satellite Earth observed ESTs (Roth et al., 1989; Dousset, 1989; Dousset and Gourmelon, 2003). However, in contrast to previous studies Nichol (2005) found that the proximity to the sea appeared to have no influence on night time EST, despite temperature differences between sea EST and coastal ESTs of up to 10°C (derived from the night-time ASTER scene). Equally, day-time ESTs from the Landsat ETM+ scene showed little variation between sea and land EST (<3°C).

In addition to studies investigating the influence of urban built density on intra-urban temperatures (e.g. Nichol (1996); Dousset and Gourmelon (2003); Nichol (2005)), urban green areas are often cited as exhibiting lower temperatures than surrounding areas. For example Nichol (2005) found daytime ESTs up to 8°C cooler in large urban green spaces (e.g. parks) than the surrounding urban areas. Similarly, Owen et al. (1998) and Dousset and Gourmelon

(2003) both found strong negative correlations between vegetation and EST in Pennsylvania and Paris respectively. Owen et al. (1998) showed that in EST pixels which were at least 25% urbanised, the vegetation-EST relationship was significant at the 95% confidence level (results of statistical testing not presented in the text). Owen et al. (1998) hypothesised that this relationship was the result of latent heat loss via evapotranspiration from vegetation, creating a cooling effect. However, Nichol (1994) and Nichol (2005) note that strong correlations between EST and vegetation are common. In studies covering both Singapore and Hong Kong (Nichol, 1994, 2005) using Landsat TM and ETM+ data, they found no meso-scale advective influence from vegetation on neighbouring ESTs. These results suggest that whilst urban green spaces are clearly cooler, they have limited influence on surrounding urban surfaces (Nichol, 1994, 2005), potentially limiting the utility of urban greening programs such as those currently employed in Singapore (Nichol, 1994) and New Jersey (Solecki et al., 2005).

2.4.4 Earth observed air temperature estimation

As previously noted (see Section 2.3.1) near surface air temperature (normally between one and two metres above the ground) is a key climatological variable (Oke, 1987) which has been used extensively for quantifying urban temperatures (Vogt et al., 1997). Thermal Earth observation data offers the possibility to derive spatially compete estimates of air temperature, removing the need to employ error prone (1-3°C) interpolation of terrestrial weather station temperature measurements (Vogt et al., 1997). For this reason a number of studies have developed empirical models to estimate near-surface air temperatures using Earth observed EST for rural and urban areas (Prihodko and Goward, 1997; Vogt et al., 1997; Czajkowski et al., 2000; Kawashima et al., 2000).

Vogt et al. (1997) used a time-series of afternoon clear sky EST scenes from the AVHRR sensor for the Andalusia region (encompassing six major Spanish cities) and a limited number of terrestrial station observations to perform

linear regression between the EST and daily maximum air temperature. The study used 148 AVHRR scenes with 3.3 km^2 windows over 31 weather stations from January to November (1992). Correlations between surface and air temperatures were found to be high (average r^2= 0.82), with a mean absolute error of 1.91°C.

Vogt et al. (1997) created a regression model to predict maximum air temperature from AVHRR EST. To evaluate the performance of the model, cross-validation was used, removing each terrestrial station in turn and comparing predicted maximum air temperature of that station with known measurements. The average mean absolute error value for all stations was found to be 2.53°C (ranging from 1.50 to 5.44°C). Individual weather station analysis revealed eight poorly performing stations, with high error values caused by their location (such as being situated on the coast or in heavily shaded locations). These stations were removed, and the model was recreated with the 24 remaining stations and showed an improvement with average r^2 increasing to 0.84, and the mean average error range decreasing (1.52 to 2.73°C). In contrast, a similar study by Czajkowski et al. (2000), which used only six AVHRR scenes and 111 terrestrial weather stations over the state of Oklahoma, United States during August 1994 (including both urban and rural regions) found a much lower correlation (r^2 = 0.64, RMSE 2.08). This indicates that potentially the number of scenes is an important factor in the strength of the regression, with a greater number of scenes providing a more robust sample of ESTs for analysis (Nichol, 2009).

Fung et al. (2009) investigated the relationship between EST and air temperatures in Hong Kong (China) using a night-time ASTER scene and 25 in-situ air and surface temperatures captured at the time of satellite overpass. The study showed a good correlation between the in-situ surface and air temperatures (r^2= 0.77). However, when the measurements were grouped by location Fung et al. (2009) found that the correlation between surface and air temperatures was worse in urban areas than those in sub-urban areas (urban r^2= 0.45, sub-urban r^2= 0.62), probably due to increased heterogeneity of surface cover and turbulent mixing in the urban areas (Oke, 1981; Prihodko and Goward,

1997; Kolokotroni and Giridharan, 2008). When deriving the EST-air temperature relationship Fung et al. (2009) created separate regression models for urban and sub-urban regions and then estimated air temperatures using EST values from the ASTER scene from urban and sub-urban pixels. A global model, derived using all measurements without grouping was also generated for comparison. Compared to in-situ air temperatures the results from the study showed that average satellite estimated air temperatures were more accurate when derived using separate urban and sub-urban models (average error 0.3°C) than using the global model which didn't account for urban/sub-urban differences (average error 0.7°C) (Fung et al., 2009).

In a related study Nichol et al. (2009) used the same ASTER scene as Fung et al. (2009) to evaluate the performance of downscaling the ASTER pixel size on the relationship between urban air temperatures and EST. Two outputs were generated from the ASTER scene during pre-processing; an EST scene using the standard ASTER temperature-emissivity separation algorithm (Gillespie et al., 1998, 1999; Nichol et al., 2009) at the native ASTER resolution of 90m, and a second EST scene processed using a down-scaling technique (see Section 2.4.2, (Nichol, 2009)) at 10m resolution. Near-surface air temperatures were captured by vehicle traverse covering 148km distance around the region within 90 minutes of satellite overpass time. A high correlation value between 10m EST and air temperatures for the same locations was found over the study area (r^2 = 0.81).

However, similar to the findings from Fung et al. (2009), Nichol (2009) found that when air temperature measurement along the car traverses were filtered by location, weaker correlations were found for urban and rural areas (Table 2.9). Nichol et al. (2009) suggests that the drastic reduction in correlation strength (r^2 reduction of up to 0.71, see Table 2.9) is because the range of rural air temperature values is much lower (4.6°C) than those captured by the ASTER EST scene over the entire rural area (11.1°C 10m, 7.1°C 90m). Potentially, this is a limitation of automobile traverse based air temperatures which are restricted spatially by the distribution of the road network meaning that the full range of rural temperatures is not being captured.

	Urban (n=8714)	**Rural (n=6100)**	**All points along traverse (n=14,814)**
EST (10m) - air temperature correlation (r^2)	0.42	0.09	0.80
EST (90m) - air temperature correlation (r^2)	0.26	0.05	0.62

Table 2.9: Correlations (r^2) between EST and air temperature along automobile traverse route for Hong Kong (source: Nichol et al. (2009)).

Furthermore, a comparison between correlation values from the 10m and 90m EST scenes in Table 2.9 shows that the added spatial resolution of the 10m scene increases the correlation between EST and air temperature on average by 0.16 (urban), 0.04 (rural). This indicates that the 10m pixel size has a closer correspondence with air temperature than the 90m pixel, capturing intra-urban spatial variation with greater accuracy (Nichol et al., 2009).

Results from studies deriving relationships between ESTs and air temperature measurements have shown high correlations between the two (e.g. r^2 = 0.81, see Table 2.9 (Nichol et al., 2009)), and using such relationships it has been possible to model air temperature from EST, with error values within the limits of the sensor's precision (Vogt et al., 1997; Fung et al., 2009). However such models are empirically based and are location, climate and season specific (Prihodko and Goward, 1997; Vogt et al., 1997; Kawashima et al., 2000; Fung et al., 2009). Voogt and Oke (2003) highlighted that knowledge of the relationship between air and satellite observed surface temperature was currently limited to empirical models which are not transferable to other regions or areas of the world. Therefore, whilst empirical models have been developed (Vogt et al., 1997; Fung et al., 2009) there is consensus in the literature that further research is required to better understand and model the surface and near surface air temperature relationship and dynamics (Voogt and Oke, 2003; Fung et al., 2009). Voogt and Oke (2003) consider that such research is fundamental to the development of models which can quantify the spatial

temperature variability of urban areas using thermal Earth observation.

2.4.5 Quantifying the urban heat island using thermal Earth observation

Thermal Earth observation has been used to study the urban heat island since the 1970's (Matson et al., 1978). Matson et al. (1978) used a single cloud-free night-time thermal scene from the Very High Resolution Radiometer (4km spatial resolution, 10.5-12.5μm) to identify the heat islands of 50 North American towns and cities using brightness temperatures. Despite using a single scene the study showed that it was possible to study the urban heat island and generate rural-urban differences using thermal Earth observed data during clear sky conditions (Matson et al., 1978). The study found urban heat island intensity in the scene (difference between rural and urban brightness temperatures) ranging from 2.6°C (Petersburg, Virginia) to 6.5°C (Louisville, Kentucky). Furthermore, Matson et al. (1978) showed that urban heat islands were visible for relatively small settlements such as Petersburg, population 40,000 (US census 1970). Matson et al. (1978) suggested that in the future such data could be used to study seasonal and meteorological trends in the UHI by making comparisons between images in a time-series.

Dousset (1989) used 84 co-registered AVHRR (bands 4: 10.3-11.3μm and 5: 11.5-12.5μm) scenes of Los Angeles from August 1984 and 1985, to investigate the micro-climates of the city and surrounding areas. Using an average of emissivity created using values from the literature, Dousset (1989) processed each of the scenes to EST, and created temporal averages of EST values from AVHRR scenes captured at the same overpass time (03:40, 07:40, 18:50, 13:55 and 15:10). Using a 120 km transect over the averaged ESTs, running perpendicular to the coast, through Los Angeles to the Mojave desert, Dousset (1989) showed that for rural regions topography and elevation were the key drivers of local temperatures (Figure 2.10). For example, during the daytime the Mojave desert (elevation ~1000m) was the warmest region (40-50°C), whilst the Santa Ana and San Gabriel mountains (inc. Mt Baded-Powell, el-

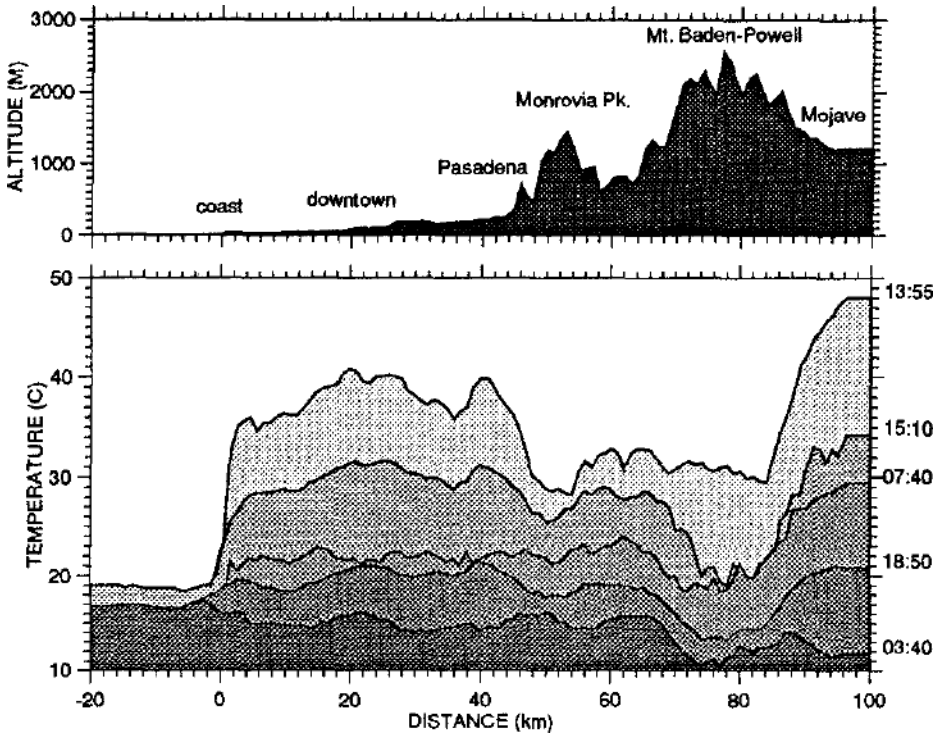


Figure 2.10: Profiles of EST and topography across the Los Angeles basin (USA) from 33°45N 118°41W to 34°27N 117°36W. Scenes captured at ~03:40 local time (11 images), ~07:40 (21), ~13:55 (15), ~15:10 (18) and 18:50 (19) (source: Dousset (1989)).

evation ~1000-2500m) were cooler (30-35°C) (Figure 2.10). During both the day and night EST values for the sea were the coolest part of the transect (15-20°C). In contrast, however, the Los Angeles basin exhibited a clear urban heat island with a maximum range between ~35-40°C during the afternoon and 15°C during the early morning (Figure 2.10)). The results from Dousset (1989) (Figure 2.10) show that the temporally averaged AVHRR EST values across the transect capture diurnal variation in the urban heat island temperatures, with rapid warming during the morning +~20°C between 07:40 and 13:55, and a cooling of ~10°C during the afternoon between 15:10 and 18:50.

As a result of initial research done by early studies in the 1970s and 1980s (e.g. Matson et al. (1978); Dousset (1989)), satellite Earth observed thermal data has been widely used to investigate the UHI effect for cities around the world (e.g. Bucharest, Budapest and Beijing; Pongracz et al. (2006); Cheval et al. (2009); Cai et al. (2011)). These studies have further developed the utility of thermal Earth observation for urban heat island analysis by deriving

the urban heat island intensity (UHII) metric using EST values (Pongracz et al., 2006; Cheval et al., 2009; Tomlinson et al., 2012; Cai et al., 2011)). This has been achieved by subtracting the EST value for a pixel in a rural location from the EST values of pixels within the urban area (Pongracz et al., 2006; Cheval et al., 2009; Tomlinson et al., 2012; Cai et al., 2011). The rural pixel is either the location of a known rural terrestrial weather station (e.g. Tomlinson et al. (2012)), or a spatial average of a buffer of pixels in rural areas surrounding the city (e.g. Pongracz et al. (2006); Cheval and Dumitrescu (2009)).

Cheval and Dumitrescu (2009) derived UHII in this manner using MODIS data for the city of Bucharest, Romania. The study used 609 scenes of the MODIS EST products (day and night, 1000m spatial resolution) to build day and night composite scenes covering the city for the month of July in the years 2000-2006. Cheval and Dumitrescu (2009) first determined the spatial extents of Bucharest's urban heat island using the ESTs, and then derived UHII within the identified extents. The extents of the daytime and nocturnal urban heat islands were then identified using the exploratory Rodionov test (a sequential application of the Student's t-test to examine significant differences between adjacent pixels (Cheval and Dumitrescu, 2009)) along 16 transects of EST spreading radially from the city centre to the urban-rural fringe (Cheval and Dumitrescu, 2009). The extent of the UHI was defined as locations along each transect where the Rodionov test showed significant differences (p=0.1) between adjacent EST values.

Cheval and Dumitrescu (2009) calculated the UHII metric using day and night EST within the identified limits using two rural buffer zones at 5 km and 10km outside the identified UHI limits. The study found the UHII derived from both 5km and 10km rural buffers was more variable, and greater during the day (night: 5km 2.2-3.7°C, 10km 2.6-3.9°C, day: 5km 2.0-5.1°C, 10km 2.5-7.1°C). These results suggest that potentially the pixels in the 5km buffer are warmer than those at 10km, and Cheval and Dumitrescu (2009) hypothesise that this is the result of different land cover types in the rural buffers. The study used the CORINE land cover database in a qualitative

manner to verify that the extent of the UHII corresponded spatially with the extents of the urban environment, though this was not extended to examine the rural buffers. Cheval and Dumitrescu (2009) propose that land cover type controls the UHI intensity and that further research using more detailed land cover information of urban areas is needed to quantify this (Voogt and Oke, 2003; Cheval and Dumitrescu, 2009).

In a similar manner to Cheval and Dumitrescu (2009), Pongracz et al. (2006) used day and night MODIS scenes (7 thermal bands 3.6 - 13.49μm) captured between 2001 and 2003 and processed to derive monthly averaged UHII for Hungary's ten largest cities (number of scenes not presented in the text). UHII was calculated using a spatial average of EST from rural pixels surrounding each city, identified as rural from a MODIS land cover image (Pongracz et al., 2006). The study then temporally averaged UHII values from all available scenes to examine UHII at the monthly scale. The study found that maximum monthly mean surface UHII occurred in all cities during daylight hours (1-4°C). There was also a notable difference between day and night UHII, with night-time UHII being 1-4 °C cooler in all cities for the summer months, indicating that daytime solar heating is the primary driver of greatest intensities (Pongracz et al., 2006). The study was also able to rank average summer intensity, finding that intensity magnitude was related to city population.

A later study by Streutker (2002) proposed a new methodology for defining the UHII using thermal Earth observation to give a better values for comparative analysis between cities or over a time-series. First a least-squares planar fit is applied to EST values for rural areas to define constant and linear components of rural temperatures, which are then subtracted from urban ESTs to generate intensities. Second, a least-squares fit to the natural logarithm of intensity values is used to fit the urban heat island intensity values to a Gaussian surface (Streutker, 2002). The Gaussian distribution provides a better characterisation of UHII for the overall city as it reduces intra-city variation (Streutker, 2002). Streutker (2002) demonstrated this method using 21 AVHRR scenes processed to EST, assuming an emissivity of 1.0 (giving

an error of 2-3°C (Streutker, 2002)). The scenes covered the city of Houston, Texas (United States) and were captured for both day and night between April 1998 and December 1999. Using the modified UHII generation methodology Streutker (2002) found a range of UHII over the series of scenes of 1.06°C-4.25°C for Houston. Additionally, Streutker (2002) showed that over the time series the urban heat island intensity was negatively correlated with rural temperature (Spearman's rank correlation coefficient -0.48 at 97% confidence level). Streutker (2002) determined that the magnitude of intensity decreased as EST increased, indicating that at greater temperatures the urban-rural difference is reduced.

The methodology for using EST values to derive a Gaussian UHII surface proposed by Streutker (2002), was used later by Streutker (2003) to perform a time-series analysis of urban heat island intensity change over time, and by Tran et al. (2006) to compare UHII between eight Asian mega cities. Streutker (2003) used two sets of AVHRR data for Houston, Texas (United States); 82 EST scenes derived from AVHRR (NOAA-9) March 1985 to February 1987, and 125 EST scenes derived from AVHRR scenes (NOAA-14) June 1999 to June 2001. All scenes were captured in the night-time between 02:00 and 05:30. Streutker (2003) showed a mean increase in UHII between the two groups of scenes (1985-1987/1999-2001) of 0.82°C (±0.10°C estimated error). Furthermore, the study found that the area of the UHII (where intensity was > 0°C) expanded 170 (± 30) km^2. Given these results, Streutker (2003) argues this is justification of the proposed method for comparing urban heat islands between points in time.

Tran et al. (2006) used the Gaussian UHII method to compare UHII values derived from MODIS scenes for Eight Asian mega cities (Tokyo, Beijing, Seoul, Shanghai, Pyongyang, Bangkok, Manila and Ho Chi Minh city). Tran et al. (2006) cites the requirement to compare UHII characteristics between cities and not absolute temperature, particularly as thermal Earth observation data for tropical cities is only available in the dry season (September-April), causing a potential bias if compared with ESTs from non-tropical cities outside of this time period. The MODIS scenes were captured between 2001 and 2002

for both day and night, and EST was generated from bands 31 and 32 (10.5-12.5 μm) using a split-window algorithm. Temporal averages of EST at the monthly level were created for UHII generation, (total number of scenes used is not presented in the text). UHII values for each month/city (day and night) were derived using the Gaussian method proposed by Streutker (2002). The average daytime UHII values over the eight cities ranged from 5.0°C (Ho Chi Minh city) to 12.0°C (Tokyo) (Table 2.10). Tran et al. (2006) also found that the daytime UHII was greater that night-time UHII for all cities (Table 2.10), with night-time UHII ranging from 2.0°C (Ho Chi Minh city) to 7.5°C (Tokyo) (Table 2.10). Tran et al. (2006) examined the relationship between day and night UHII and population size for each city to see if there was a correlation between population size and UHII. Tran et al. (2006) found that daytime UHII values were positively correlated to population size (r = 0.93, at significance level p<0.1%), and night-time UHII, but with weaker relationship (r = 0.75, at significance level p < 0.1%). Tran et al. (2006) states that these result indicate significant impacts of urban growth on the UHII in Asia, as projections of population increase could cause an increase in UHI.

	UHI intensity	
City	**Day**	**Night**
Tokyo	12.0	7.5
Beijing	10.0	5.5
Shanghai	7.0	3.5
Seoul	8.0	4.5
Pyongyang	4.0	3.0
Bangkok	8.0	3.0
Manila	7.0	2.0
Ho Chi Minh City	5.0	2.0

Table 2.10: UHII for eight selected Asian mega cities using MODIS EST and the Gaussian UHII method proposed by Streutker (2002) (source: Tran et al. (2006)).

2.4.6 Thermal Earth observation of heatwaves

Currently, heat-health warning systems (HHWS) don't quantify spatial variability in exposure over the urban surface (Kovats and Ebi, 2006; Dousset

et al., 2011), although it has been shown that there are large spatial variations in vulnerability and exposure dependent on socio-economic variables as well as urban temperatures (Harlan et al., 2006; Gaffin et al., 2008). Due to the poor spatial coverage of terrestrial weather stations a limited number of studies are now utilising Earth observed temperature data-sets to measure spatial and temporal variability in urban temperatures during previous heatwaves to assess spatially the potential exposure during extreme temperature events (Cheval and Dumitrescu, 2009; Tomlinson et al., 2012; Dousset et al., 2011).

To investigate the utility of satellite Earth observation to capture heatwave dynamics Dousset et al. (2011) studied the 2003 heatwave of Paris (France) which lasted 9 days and resulted in 4,867 excess heat-related deaths in the Paris metropolitan area (Dousset et al., 2011). Dousset et al. (2011) used 61 day and night AVHRR scenes (July-August) processed to EST to identify relationships between temperature and spatio-temporal variation in heat-related mortality.

Using public health records the study identified 482 addresses in the Paris metropolitan area, half of which had experienced mortality as a result of the heatwave and the remaining selected to act as a control. Over the series of AVHRR scenes 29,000 individual EST measurements were extracted, corresponding to the locations of the identified addresses. The study used an odds ratio to assess the risk of death as a function of the temperature at each address. An odds ratio is a relative measure of risk that indicates the probability of someone exposed to increased levels of a hazard (e.g. heat) developing a detrimental outcome (e.g. mortality), as compared to someone with less or no exposure to the hazard (Dousset et al., 2011). The odds ratio was calculated using a mean average of minimum nocturnal EST for the seven days preceding the date of death. The results showed that an increase in EST of 0.5°C between control and case addresses lead to an odds ration of 2.2, indicating that the risk of death at the warmer addresses was twice as high (Dousset et al., 2011).

However, this relationship was only found for night-time temperatures, suggesting that night-time temperatures were a driver for increased mortality during the heatwave. The study showed strong daytime and nocturnal heat islands during the heatwave (maximum: 8°C night-time, 11°C daytime). Dousset et al. (2011) suggest that the urban heat island had a significant contribution to the heatwave effect, absorbing heat during the day and progressively increasing re-radiated nocturnal temperatures through the nine day event causing a ~7°C rise in minimum temperatures in the city between 1st-12th July.

The study also found that the distribution of temporally averaged afternoon EST for 2003 (scenes between 12:00-15:00) correlated well with building density, and that urban green spaces such as parks had a cooling effect, with a 0.2°C decrease in EST per percentage increase in NDVI (NDVI was a temporal mean average from 4rd-13th August, scaled as a percentage). Dousset et al. (2011) highlight that the use of thermal satellite Earth observation allows the same methods to be used for any city around the globe without the constraint of acquiring terrestrial air temperature observations, providing that ancillary mortality data are available. However, the finding that nocturnal temperatures were the primary driver for increased mortality in Paris are in direct contrast with the results of terrestrial air-temperature based studies, such as Harlan et al. (2006). Harlan et al. (2006) showed that for Phoenix (Arizona, USA) increased mortality as a result of extreme temperatures occurred during the day due to amplification of daytime temperatures and population overexertion during extreme temperature events. However, unlike Paris and other European cities, Phoenix's nocturnal mortality risk is largely mitigated by air conditioning (Harlan et al., 2006).

Tomlinson et al. (2012) studied the response of the nocturnal heat island intensity to a heatwave event in the city of Birmingham (United Kingdom). Using 63 MODIS images with an overpass time of 01:30 for summers between 2003 and 2009 (including the heatwave of July 2006) the UHII was calculated to generate a series of night-time UHII images. During non-heatwave summers the maximum UHII ranged from 1.8°C (calm weather conditions)

to 3.09°C (unsettled weather conditions). The results revealed that during the heatwave event maximum UHII increased to 4.88°C, supporting the findings of Dousset et al. (2011), that the nocturnal urban heat island intensity increases during a heatwave event (Tomlinson et al., 2012). Furthermore, Tomlinson et al. (2012) showed that the location of maximum UHII and urban 'cool-islands' did not change during the heatwave, indicating that the heatwave did not result in a change to the spatial distribution of the heat island.

Cheval et al. (2009) used fully diurnal (both day and night data) air temperature from terrestrial observations and MODIS imagery to generate UHII for Bucharest (Romania) during the July 2007 heatwave that affected the region. Urban heat island intensity was generated using the method described above in Cheval and Dumitrescu (2009) using a spatially averaged rural temperature. As in the case of Tomlinson et al. (2012) no spatial expansion of the urban heat island outside the already established limits at the rural-urban boundary were observed (Cheval and Dumitrescu, 2009). However, in contrast to Tomlinson et al. (2012) the UHII did not appear to exhibit increased values during the heatwave month, although this was not quantified statistically, suggesting that further research is required to quantify whether there is a consistent increase in intensity during heatwaves and whether this is captured by the UHII.

2.5 Discussion

It is now widely accepted that the urban heat island exacerbates heatwave exposure by increasing daytime (Harlan et al., 2006) and night-time temperatures, reducing urban cooling (Kovats and Hajat, 2008; Dousset et al., 2011). A number of studies have quantified the increase in mortality found in cities during heatwave events (Baker et al., 2002; Johnson et al., 2005; Harlan et al., 2006; Gaffin et al., 2008), and as such heatwave impacts are forming core themes in future cities climate change adaptation and mitiga-

tion strategies such as ARC3 (Solecki et al., 2005; Rosenzweig et al., 2005; Mehrotra et al., 2009; Barata et al., 2011; Mehrotra et al., 2011; Rosenzweig, 2011; Rosenzweig et al., 2011). Furthermore, studies have shown that the urban heat island is not uniformly distributed across the city (Kolokotroni and Giridharan, 2008; Cheval and Dumitrescu, 2009; Tomlinson et al., 2012), indicating the increased exposure to the heat hazard caused by the UHI is spatially variable (Dousset et al., 2011). Thermal Earth observation has proven itself in this regard, demonstrating its ability to capture spatial variability in intra-urban temperatures (Nichol, 2005; Cheval and Dumitrescu, 2009), information which is key to identifying areas of highest exposure for future heatwave adaptation and mitigation options (Dousset et al., 2011).

The provision of spatially-complete ESTs to quantify intra-urban temperatures has also enabled studies to examine the influence of different urban surface characteristics on intra-urban temperatures across the entire urban area (Nichol, 1994, 2005). Such information was not previously achievable with spatially discrete air temperature measurements (Nichol, 1994, 2005). For example, using Landsat and ASTER imagery Nichol (1994) and Nichol (2005) showed that cooling effects of the ocean in coastal cities and urban green-spaces appeared to have little influence on adjacent (<1km) densely urbanised areas. These findings are in direct contrast with existing theories regarding the cooling influence of large water bodies and urban green space on intra-urban air temperatures (Oke, 1987; Eliasson and Svensson, 2003; Harlan et al., 2006).

Given the wide use of near-surface air temperatures in urban climate research (Oke, 1987; Eliasson and Svensson, 2003; Harlan et al., 2006), a number of studies have used empirically derived models to estimate spatially complete air temperatures from thermal Earth observation (Vogt et al., 1997; Fung et al., 2009; Nichol et al., 2009). Results indicate that empirical models can provide good estimates for air temperature (Vogt et al., 1997; Nichol et al., 2009), and that ESTs from higher spatial resolution sensors provide a better relationship with air temperature (Nichol et al., 2009). However, the literature shows that there are two disadvantages to the use of such models; firstly

that deriving empirical relationships using just urban (not including rural areas) provides poor correlation between EST and air temperatures (Nichol et al., 2009), and second, because of their empirical nature, the use of such models is spatially and temporally constrained to the region and time under which the relationship was derived. Therefore there is a recognition in the literature that further research is required to better model and understand EST-air temperature dynamics (Voogt and Oke, 2003).

Traditionally, studies have used the UHII metric to quantify the heat island effect, and the metric has been successfully generated from long time series data to test for the existence of heat island intensification during the last century (Lee, 1992; Wilby, 2003; Gaffin et al., 2008; Jones et al., 2008). Furthermore, the metric has been generated using thermal Earth observed ESTs using the difference between urban and rural pixels on a scene-by-scene or temporal scene average basis to generate intensity maps for different cities (Streutker, 2002; Pongracz et al., 2006; Cheval and Dumitrescu, 2009; Cheval et al., 2009). Results from studies using EST derived UHII values, in conjunction with a limited number of studies using spatially-dense terrestrial air temperature (e.g. Kolokotroni and Giridharan (2008)), have shown that contrary to previous characterisations of the urban heat island (Oke, 1987) the city centre does not necessarily exhibit the greatest heat island intensity (Harlan et al., 2006; Tran et al., 2006; Kolokotroni and Giridharan, 2008). Furthermore, the same studies have shown that maximum UHII does not always occur during the night (Harlan et al., 2006; Tran et al., 2006; Kolokotroni and Giridharan, 2008), as is generally accepted (Oke, 1987).

This review has also shown that there is a discrepancy in the literature investigating urban heat island intensification of heatwaves. The majority of studies choose to quantify heatwave events and their impacts in terms of absolute temperature or thermal stress indicators (Baker et al., 2002; Harlan et al., 2006) and few if any assess directly the relationship between the heat island and heatwave temperatures using the intensity metric. Furthermore, the literature shows a distinctly limited number of studies which have derived the UHII metric using thermal Earth observation for spatial analysis

of intra-urban temperature dynamics during a heatwave event (Cheval et al., 2009; Tomlinson et al., 2012; Dousset et al., 2011).

The limited number of studies which have employed the UHII metric for heat island analysis have not shown a consensus on whether urban heat island intensity increases during a heatwave, and if such an increase exists, whether this is captured by the UHII (Cheval et al., 2009; Tomlinson et al., 2012; Dousset et al., 2011). These results, and the limited number of studies employing the UHII metric for heatwave analysis indicate that further research is required to examine whether intensity as quantified by EST-UHII increases during heatwave events, and so can be used to quantify the spatial variability of exposure to the heat hazard in cities during heatwaves.

2.6 Conclusions

In conclusion, to better understand intra-urban temperature variability in cities during heatwaves further research needs to be undertaken to assess the response of temperatures and the UHII metric to heatwave events. Critically, this needs to be undertaken using thermal Earth observation data integrated with terrestrial air temperature observations over a long-time series to ensure complete spatial and temporal coverage across urban areas. To meet the aim and objectives stated in Section 1.3 the research presented in this thesis attempts to address these research challenges in two ways. First, a long-time series of thermal Earth observation and terrestrial air temperature measurements will be used to investigate the ability of empirical models to generate spatially-complete air temperature across a city, valid for heatwave and non-heatwave years. Second, the utility of thermal Earth observation to quantify the magnitude of a heatwave summer using EST and the UHII metric will be evaluated.

Chapter 3

Data acquisition and pre-processing

This chapter presents the data-sets which are employed in this thesis to quantify the intra-urban temperature dynamics of London. Two data-sets were used; the first was the Met Office Integrated Data Archive System (MIDAS) terrestrial hourly and daily air temperature records from United Kingdom Met Office weather stations; the second was a time-series of satellite images from the Advanced Very High Resolution Radiometer (AVHRR) satellite sensor covering summer months (May-October) in the UK from 1985 to 2008. To enable spatio-temporal integration of the two data-sets a spatial database was designed and constructed for the MIDAS terrestrial weather station temperatures. The processes of database creation and data-filtering for the MIDAS data are discussed in Section 3.1. Furthermore, a number of pre-processing steps had to be performed on the AVHRR data to derive estimated surface temperatures (EST). A description of these steps and justification of the chosen AVHRR pre-processing methodology are discussed in Section 3.2. Section 3.3 provides a summary of the pre-processing steps for both data-sets in preparation for analysis.

3.1 MIDAS PostGIS database

This project utilised hourly screen-level air temperature observations and daily (minimum) surface temperature measurements recorded at UK Met Office (UKMO) weather stations. This data is part of the MIDAS dataset provided by the British Atmospheric Data Centre (BADC). A selection of the MIDAS data covering the same time-series as the AVHRR data (see Section 3.2) from 1985 to 2008 for all UK weather stations was downloaded from the BADC. In total the downloaded data-set exceeded fifty million records.

To help manage, store and analyse this large data-set a spatial PostGIS database was created for the MIDAS data. PostGIS spatially-enables the popular PostgreSQL relational database by allowing the encoding of geometry within database tables. The aim of the database was to store the locations of weather stations and their measurements as a series of uniquely identifiable records. In order to construct a coherent dataset within the database the following objectives were defined:

1. Remove null, incorrect and duplicate values from the data.
2. Identify gaps in the data, either inherent in the original data or as a result of removal due to step 1.
3. Create SQL statements to efficiently select data from the database for the required weather stations and time periods required for analysis.

3.1.1 The MIDAS dataset

Three data files were download from the BADC in comma separated value (CSV) text format; station_id_list_.txt, midas_wxhrly_'yyyymm'-'yyyymm'.txt, midas_tempdrnl_'yyyymm'-'yyyymm'.txt. The first file consisted of a listing of weather stations within the United Kingdom. The second group of files contained hourly screen-level air temperature measurements for all stations between 1985-01-01 and 2008-12-31 (one file per year), the third contained

daily (minimum) surface temperature measurements for all stations between 1985-01-01 and 2008-12-31. The total volume of raw files was approximately 15 gigabytes, containing 52,696,386 hourly air temperature measurements, 7,109,751 daily surface temperatures and 50,043 station records. For clarity the first file is hereby referred to as 'station data', the second as 'hourly air data' and the third as 'daily surface data'. Figure 3.1 shows the pre-processing flow-line for MIDAS stations and hourly air data from raw files to database tables, which was divided into three main stages. First, hourly air data and the station data underwent preliminary data-cleaning and filtering (see Section 3.1.2, and Section 3.1.3). Second, the data were loaded into the MIDAS spatial database, creating a new database table for each (see Figure 3.1). Third, further data-cleaning and filtering was carried out in the database to remove erroneous values and duplicate measurements to produce the final tables of weather station attributes and hourly air data (see Sections 3.1.4, 3.1.5 and 3.1.6). Figure 3.1 doesn't show the pre-processing methodology for the daily surface data as this was the same as the hourly air temperature pre-processing, repeated to create a new surface temperatures table in the database.

Screen-level air temperature is measured inside a Stevenson screen to protect the instruments, at a standard height of 1.25m in the UK (UKMO, 2012). Traditionally, air temperature measurements were made by observation of the thermometer inside the screen once an hour, but during the 1970's and 1980's nearly all stations were automated and now use an electrical resistance thermometer to record air temperature (UKMO, 2012).

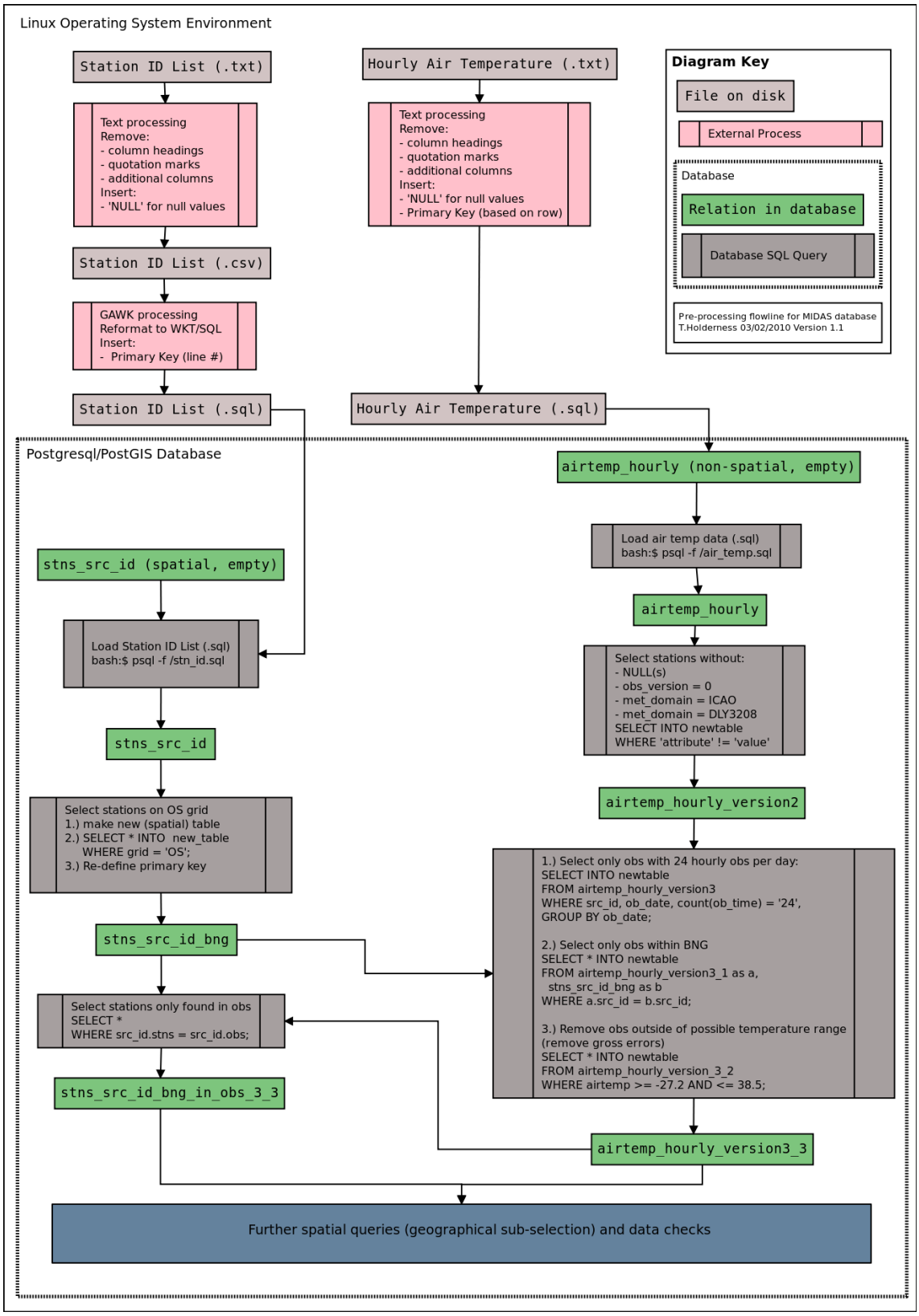


Figure 3.1: Flow-line showing the pre-processing methodologies for the MIDAS terrestrial temperature data including creation of the PostGIS spatial database.

Minimum surface temperatures are the lowest temperature recorded overnight by a thermometer freely exposed to the sky with its bulb just touching the tip of short grass (25-50mm above the ground) or set in the centre, and just touching a concrete slab (UKMO, 2012). Figure 3.2 shows a typical arrangement of meteorological instruments at the St James's Park weather station, where the Stevenson screen and concrete slab can be clearly seen. Since 1982 all temperatures in the MIDAS data-set are stored with a precision of 0.1°C. Instruments are calibrated by the Met-Office ensuring a measurement accuracy of 0.2°C and instrument corrections derived from regular calibration are applied to temperatures to maintain this level of accuracy (UKMO, 2012).

Figure 3.2: Weather station instruments at St James's Park, London.

3.1.2 Pre-processing the MIDAS station information

The station data contains the location and meta-data attributes for Met Office weather stations within the UK, Northern Ireland and the Channel Islands. The station location is defined by latitude and longitude in World Geodetic System 1984 (WGS84) as well as Eastings and Northings in a variety of projected coordinate reference systems including the British National Grid (BNG). Within the station data file there were 50,043 rows of data, corresponding to a total of 16,615 individual stations. The total number of sta-

tions is lower than the file content as a number of stations (with the same id number and location) have duplicate listings as they appear in multiple meteorological networks.

One of the considerations during pre-processing of the MIDAS data was to create a file in CSV format which was compatible with the PostgreSQL database software so that the MIDAS files could be loaded directly into the database. It is possible to load CSV data files into a PostgreSQL database using the 'COPY FROM' function, but this method doesn't support the loading of spatial data into a PostGIS table. An alternative method is to create an SQL file which contains insert commands for each tuple of data to be entered into the database (one tuple per row in the CSV file). Station location coordinates may be included in the SQL command using the 'ST_GeomFromText()' PostGIS function, which takes location in Well Known Text (WKT) syntax as defined by the Open Geospatial Consortium and encodes the values in the database as geometry.

To ensure that the data conformed to the SQL standard the first stage of preprocessing was to use a text editor to remove column headings, extraneous metadata, quotation marks and insert the term 'NULL' for missing values, the output of which was a new text file in CSV format. In order to convert the CSV text file to SQL a parser program was written in Bash to handle the conversion. Appendix C provides the source code for the parser script. In addition to reformatting, the parser also removed fields not pertaining to temperature measurement (Table 3.1) and added a primary key for each record in the station file. For the station data the primary key was derived by the parser from the line number in the station data file (stations were listed in alphabetical order, one per line). An empty database table to hold the station location and attributes was created and the SQL file was loaded into the database from the shell using the PSQL command. Due to the mix of coordinate systems within the station data file, the station data was initially geocoded in the database using the WGS84 latitude and longitude attributes. Once entered into the database, stations outside the British mainland (not having BNG coordinates) were removed.

Short name	Description
SRC_ID	Station identification integer
SRC_NAME	Station name
ID_T	ID Type (station type, e.g. wind)
ID	Identification number (ID type number)
MET_DOMA	Met domain Source (network)
SRC_CAP_BG	Source capability begin date
SRC_CAP_EN	Source capability end date
LOC_	Shorthand for Area (e.g. county name)
POST_CODE	Post code
HIGH_PRCN_LAT	High precision latitude
HIGH_PRCN_LON	High precision longitude
GRID	Grid Coordinate Reference System
EAST_GRID_REF	Eastings
NORTH_GRID_REF	Northings
ELEVATION	Elevation
HYDR_AREA_ID	*Hydrological Area ID*
DRAI	*Drainage Area*
SRC_BGN_DA	Source begin date
SRC_END_DA	Source end date

Table 3.1: UK Met Office weather station attribute names and descriptions from the MIDAS station data. *Note: italicised fields correspond to removed attributes not pertaining to temperature measurement.*

3.1.3 Pre-processing the MIDAS temperature measurements

The hourly and daily surface data do not contain any geometry, therefore the PostgreSQL function 'copy from file' could be used to load the CSV file directly into new database tables. However, the hourly and daily CSV data files required formatting before they could be loaded into the database. A new parser script was created for the hourly and daily temperature data (Appendix D) a summary of the script operations is as follows:

1. Select the following fields: "date/time, id, id_type, met_domain, version_num, src_id and air_temp"
2. Remove trailing commas
3. Insert a comma between time and date

4. Add a primary key attribute based on line number

5. Create one file for all observations

Table 3.2 shows the key attributes relating to air temperature measurements (UKMO, 2012) selected from hourly air data to be entered into the database. The table was named 'airtemp_hourly_uk_19852008_clean_input' and was considered version 1.0 of the hourly air data. Subsequent tables created from this data were assigned version numbers (e.g. 'airtemp_hourly_uk_ 19852008_2_1' signifying version 2.1 of that table). The same process was then repeated for the daily surface temperature data (Table 3.3).

Short name (as selected for database)	**Description**
OB_DATE	Observation date
OB_TIME	Observation time
ID	Network ID Number
ID_TYPE	Network ID (e.g. WMO)
MET_DOMAIN_NAME	Message type of observation
VERSION_NUM	Measurement Version Number
SRC_ID	Station ID
AIR_TEMP	Screen-level air temperature (± 0.1°C)

Table 3.2: MIDAS hourly air temperature data table attributes.

Short name (as selected for database)	Description
OB_DATE_END	Observation date end
OB_END_TIME	Observation end time
OB_HOUR_COUNT	Observation hour count (12 or 24 hours)
ID	Network ID Number
ID_TYPE	Network ID (e.g. WMO)
MET_DOMAIN_NAME	Message type of observation
VERSION_NUM	Measurement Version Number
SRC_ID	Station ID
MIN_GRASS_TEMP	Minimum grass temperature ($\pm 0.1^{\circ}$C)
MIN_CONC_TEMP	Minimum concrete temperature ($\pm 0.1^{\circ}$C)
MIN_GRASS_TEMP_Q	Minimum grass temperature quality control code
MIN_CONC_TEMP_Q	Minimum grass temperature quality control code

Table 3.3: MIDAS daily grass and concrete surface temperature data table attributes.

3.1.4 Removing null and erroneous values

Once all three datasets (hourly and daily surface data, and station information) were loaded into the database, further pre-processing was completed within the database environment. The MIDAS data contained measurements where the air temperature field was null which had not been detected during the pre-processing stages. Therefore measurements with null values were removed from the hourly air temperature table using an SQL script (see Appendix G). The MIDAS temperature measurements also include a measurement version number attribute (see Table 3.2, 'VERSION_NUM') which is used to indicate quality of the measurement and is not related to the aforementioned table version number. A measurement version number of zero indicates an incorrect observation, which were also removed from the hourly air temperature data. Table 3.4 shows the number of air temperature measurements in the table and those removed at this stage. Appendix E documents the SQL code used to select and remove measurements from the database.

Description	Count of Tuples	As Percentage Of Input Records (%)
Number of input records	52,696,386	100
Total number of NULL values	6,158,127	11.69
Total number of version '0' values	883,407	1.676
Count of records with both NULL and '0' values	21,827	0.041
Number of 'clean records' (i.e. no NULL or '0')	45,676,679	86.68

Table 3.4: Number of null and erroneous measurements removed from MIDAS air temperature table during pre-processing.

3.1.5 Duplicate Observations

From inspection of the data it was found that hourly air temperature records contained duplicate measurements. In some cases duplicates were an exact copy (i.e. all data in duplicate rows were the same). There was also a second type of duplicate where the station ('SRC_ID'), time ('OB_TIME') and temperature ('AIR_TEMP') attributes are all duplicated, but meteorological network ('MET_DOMAIN_NAME') attributes were different. Further interrogation of the data indicated that this frequently occurred with stations that had observations with ID type 'ICAO' International Civil Aviation Authority (UK stations part of the European aviation meteorology network), which constitute the same temperature measurement but in integer form (i.e. truncated to whole numbers). Duplication of observations may also exist due to human error not identified by the version numbering system. For example hourly air data from station 613, Benson, Oxfordshire is currently received by the UK Met Office on a postcard from a private observer and data transcription may introduce additional errors.

It was important to remove duplicates from the database in order to ensure that daily averages from each observation location were not biased as a result of having more than 24 hourly observations per diurnal period. However, due to the volume of records and number of stations (1,118 stations in the GLA which operated at some point between 1985-2008) it was not possible to analyse data for each station to identify which observations should be removed to

correct measurement duplication. Therefore, a data-wide filtering approach was employed to remove duplicate measurements. Inspection of duplicated data showed that observations from some meteorological networks were more frequently part of duplicated data than others (e.g. from the ICAO meteorological network) and so measurements from these networks were removed from the hourly air temperature table. In order to maintain data flow-line transparency during the removal of data, records were not deleted, but instead a new copy of the table (with an incremental version number) was created with only the records required (i.e. without those to be removed). In this way no data was actually erased from the database and tables could be rolled back to a previous version if required.

Due to the aforementioned large number of duplicated records with ID type 'ICAO', all measurements of this type were removed (around 11% of the data). In addition, those with meteorological domain name 'DLY3208' (hourly climate return) also created a high number of duplicates and so were also removed. Measurements which contributed to diurnal periods with fewer than 24 hourly observations were also removed. At this point, measurements from outside the British National Grid were also removed. Finally, any observations that contributed more than 24 records per day which were not remedied by removing duplicates were also deleted. Table 3.5 shows the numbers of records for each attribute that were deleted. The rigorous removal of data was intended to provide a data-set which only contained error free, complete daily hourly air temperature observations from UK weather stations. This meant that all days in the database contained 24 error free hourly air temperature measurements. As daily surface temperature data were only used for specific dates and times during the AVHRR pre-processing checks for cloud contamination (see Section 3.2.8) the same data-wide duplicate removal was not required. Instead, surface temperature measurements were filtered for duplicates at the time of selection for comparison with AVHRR data (see Section 3.2.8).

Description	Records Removed	Records Remaining in Database
Number of records in original database table	-	45,676,679
Number of ICAO records	5,123,145	40,553,534
Number of DLY3208 records	3,393,431	37,190,103
Number of records < 24 obs per day	5,981,503	31,178,600
Number of records outside of BNG	3,060,800	28,117,800
Number of records not 24 obs per day	3,090,648	25,027,152

Table 3.5: Duplicate and erroneous hourly air temperature records removed from the MIDAS data.

3.1.6 Data Checking

Due to missing hourly records and duplicate values in the raw data the final hourly air temperature table contained gaps in the time-series for each station. As all days with greater or fewer than 24 measurements per day were removed, the gap unit time length is one day (i.e. only whole days are missing, therefore seven sequential gaps would equal one week). To identify gaps in the data it was necessary to order the data in ascending date order and compare the difference in date between adjacent data records in the observations table. The Python programming language contains a datetime module, with which it is possible to perform unary operations on Gregorian dates, for example `02/06/2003 - 01/06/2003 = 1 day`.

A Python script was written to use the datetime module to test for differences between input dates (see Appendix F). To pass data from the database to the Python script as a series of comma separated values, a C utility called CSQL was written (see Section 4.3 and Appendix G). This utility used the libpq library (an application programming interface to PostgreSQL databases) to pass data to and from the database in memory via the standard input/output interface in the Linux operating system, and is discussed further in Section 4.3. The CSQL utility passed the hourly air temperature records from the database table to the Python script, which in turn calculated the difference in time between adjacent measurement and reported gaps in the time series. The generated reports of time-series gaps were used for analysis of missing

data by station during station selection for analysis (see Section 5.2).

3.1.7 Summary of MIDAS pre-processing

After pre-processing of the MIDAS data, the PostgreSQL database contained three tables representing; 4,313 Met-Office weather stations on the British mainland, 25,027,257 hourly air temperature records and 7,068,957 daily surface temperature measurements. The hourly and daily measurement tables were linked to the station location using the station ID ('SRC_ID') field, as provided in the MIDAS data. A one to many relationship between each station and its measurements meant that air and surface temperatures could be queried by location using the geometry of the station location as encoded by the PostGIS extension. Furthermore, the date and time fields in the temperature tables (see Table 3.2 and Table 3.3) meant the measurements could also be selected for analysis by date and time. These attributes were subsequently used to perform spatio-temporal pairing of air temperatures and AVHRR ESTs for analysis (see Section 5.2.3).

3.2 AVHRR data

3.2.1 Introduction

As discussed in Section 2.4 the thermal characteristics of urban areas have been studied using a range of sensors from Earth observation satellites, including the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) (Golden, 2004; Kato and Yamaguchi, 2005; Lu and Weng, 2006; Kato and Yamaguchi, 2007; Cai et al., 2011), the Landsat programme (Nichol, 1996; Lo and Quattrochi, 2003; Nichol, 2003; Weng et al., 2004; Tran et al., 2006; Liang and Weng, 2008; Cai et al., 2011), the Advanced Very High Resolution Radiometer (AVHRR) (Roth et al., 1989; Gallo et al., 1993; Lee, 1993; Streutker, 2002, 2003; Voogt and Oke, 2003; Stathopoulou and Cartalis, 2009),

and the Moderate-resolution Imaging Spectroradiometer (MODIS) (Jin et al., 2005; Tran et al., 2006; Pongracz et al., 2006; Cheval and Dumitrescu, 2009; Cheval et al., 2009; Tomlinson et al., 2012).

Critically, analysis of intra-urban temperatures needs to be performed with a suitable time series of imagery acquired over an appropriate time-period in order to be able to realistically characterise the urban temperature property of interest (Nichol, 2003; Kato and Yamaguchi, 2005). For example, studies employing MODIS and Landsat TM data to analyse the temporal variations of UHI between eight 'mega' cities in Asia (Tran et al., 2006) highlighted the importance of having a sufficient time-series of data to be able to not only derive metrics that are insensitive to local diurnal fluctuations, such as meteorological conditions, but also to capture the long term spatial and temporal temperature dynamics required for monitoring (Tran et al., 2006).

The AVHRR sensor offers a long-time series (1978 to the present day) of data captured in the visible, near, shortwave and thermal infrared regions on the electromagnetic spectrum (Cracknell, 1997). The long-time series of data has been used to study intra-urban temperature dynamics over single and multiple years (Lee, 1993; Stathopoulou and Cartalis, 2009; Dousset et al., 2011). The sensor has a spatial resolution of 1.1Km at nadir and an across-track swath width of 2399Km. The AVHRR sensor was first deployed on the Television Infrared Observation Satellite (TIROS-N) in 1978. The AVHRR sensor is currently operational on the National Ocean and Atmospheric Administration (NOAA) series of Polar Orbiting Environmental Satellites (POES) which began operation in 1979 and continue to the present day. Deployment of the AVHRR sensor on multiple satellites provides up to four overpasses of the sensor per day, meaning that AVHRR data can be used for temporal analysis at the sub-diurnal level.

The first AVHRR sensor captured imagery in just four bands (Table 3.6) which was improved to five by the launch of NOAA-7 in 1981, with the addition of an extra thermal infrared band (Cracknell, 1997). More recently, the latest version of the sensor (first carried by NOAA-15 in 1998) has six bands, with

Band	TIROS-N	NOAA 6, 8, 10	NOAA 7, 9, 11, 12, 14	NOAA 15 - Present
1	0.55-0.9 μm	0.58-0.68 μm	0.58-0.68 μm	0.5-0.68 μm
2	0.725-1.1 μm	0.725-1.1 μm	0.725-1.1 μm	0.73-1.1 μm
3A	-	-	-	1.58-1.64 μm
3B	3.55-3.93 μm	3.55-3.93 μm	3.55-3.93 μm	3.55-3.93 μm
4	10.5-11.5 μm	10.5-11.5 μm	10.5-11.3 μm	10.3-11.3 μm
5	Band 4 repeat		3.55-3.93 μm	11.5-12.5 μm

Table 3.6: Spectral characteristics of the AVHRR sensors.

the addition of a mid-infrared band (3A) as an alternative to the existing mid-infrared band 3B, which is primarily used for night time cloud and sea surface temperature mapping (Cracknell, 1997).

With respect to its provision of a long time-series of data and multiple daily overpasses this study chose to use data from daytime AVHRR scenes to provide a series of estimated surface temperatures for London. The choice of daytime scenes was taken for a number of reasons. Firstly, recent work on the UHII of London (a key metric within this project) has revealed that maximum air temperature UHII in London occurs in the daytime (Kolokotroni and Giridharan, 2008). Thus, in Chapter 6 the study evaluates the utility of AVHRR data to express this feature. Secondly, a significantly greater number of scenes were available for daytime hours (1718 scenes) compared to night-time (698). In relation to this point, the check for cloud contamination (Section 3.2.8) employed is more reliable when using daytime images than night-time data. Finally, the use of daytime images allowed the study to employ a surface emissivity correction procedure based on the Normalised Difference Vegetation Index (NDVI) using AVHRR bands 1 and 2 (Van De Griend and Owe, 1993). Night-time image data would not allow such a correction procedure to be employed.

3.2.2 Data acquisition

For this project an application was made to the Dundee Satellite Receiving Station (DSS), a Natural Environment Research Council (NERC) facility which receives and archives AVHRR data. The application was accepted and DSS agreed to provide access to their entire archive of scenes covering the years 1985 to 2008. Figure 3.3 shows the data ordering and pre-processing steps undertaken to derive estimated surface temperature for the Greater London area. However, prior to submitting the final order for data it was necessary to consider the data format, georeferencing, and radiometric correction options of data for the final data product to be delivered by DSS. A discussion of these considerations is presented in Sections 3.2.3, 3.2.4 and 3.2.5 respectively.

The final data order was created using the DSS Pass Database which contains AVHRR scene meta-data, and which was used to create a PostGIS AVHRR scene meta-data database (Figure 3.3, see Section 3.2.6 and Section 3.2.7). Once the finalised data order was received preliminary processing of the AVHRR data was undertaken (Figure 3.3) to match the scenes to the study area and perform an additional check for cloud contamination (Section 3.2.8). The final stage of AVHRR pre-processing was to implement corrections for atmospheric attenuation and emissivity to derive estimates of surface temperature (Figure 3.3). Sections 3.2.9 and 3.2.10 present an overview of methods for atmospheric correction of visible and thermal imagery from the literature. Section 3.2.11 discusses methods for correction of surface emissivity, and Section 3.2.12 presents an operational approach for atmospheric and emissivity correction of AVHRR thermal bands used in this thesis.

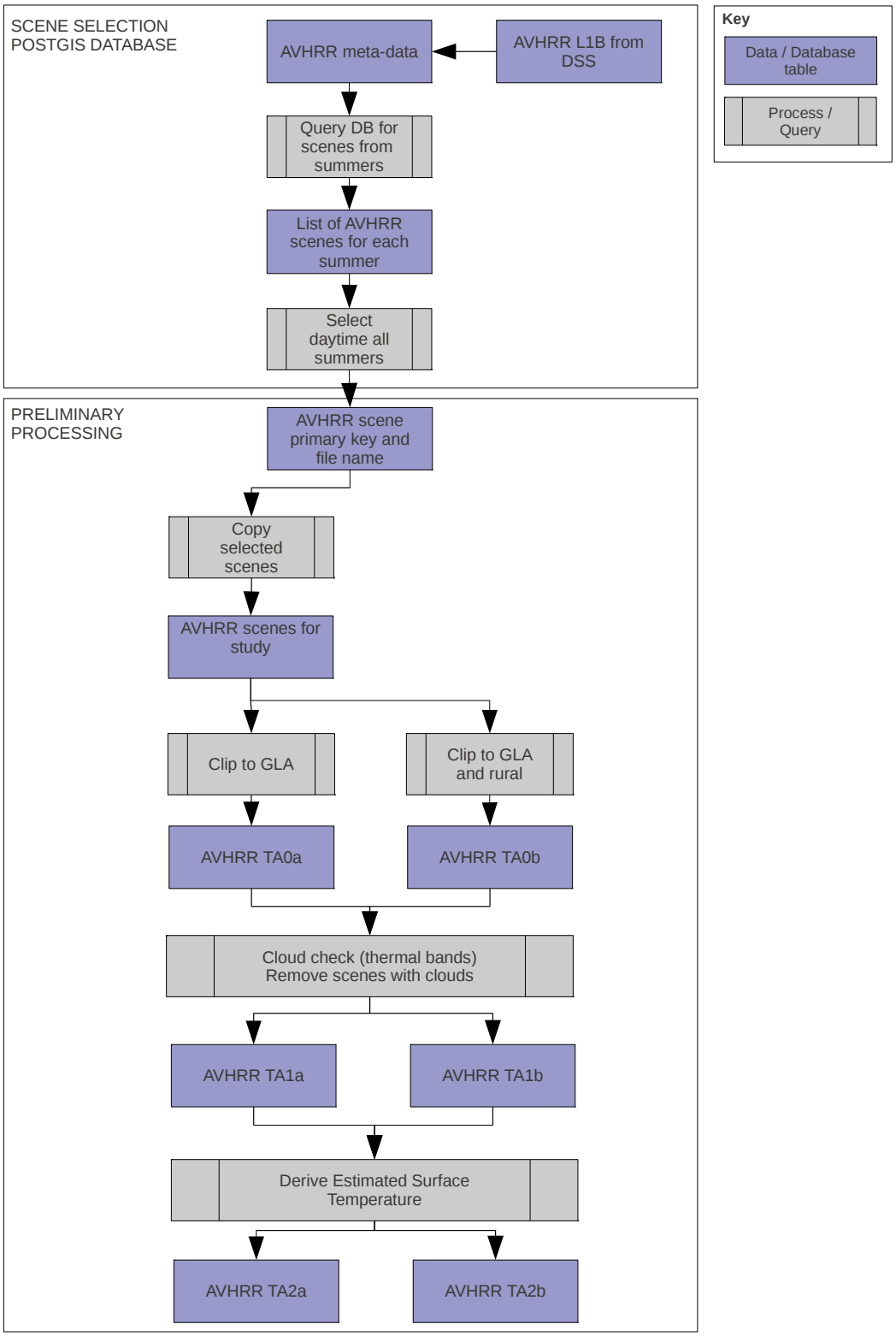


Figure 3.3: Flow-diagram showing the pre-processing to derive EST from the time-series of AVHRR scenes.

3.2.3 Data format and scene extraction

Prior to the finalised data order, a sample of AVHRR data was provided by the Dundee Satellite Receiving Station (DSS). The sample was provided as NOAA level 1B (N1B) 10-bit packed (compressed) binary format, as defined in the NOAA Polar Orbiter Data User's Guide (NOAA, 2009). This format is used by Satellite Service Branches (SSB) such as DSS to archive AVHRR imagery. The 10-bit packed compression minimises disk space needed for image storage by removing space between data elements. The compression works by interleaving the pixel data for each channel and storing three 10-bit samples in four bytes (with the first two bits of each four-byte group as zero to create a 32-bit line). Due to the compressed nature of the data, each image must be unpacked before it can be used.

DSS provided the source code of the "nibx.c" software which is used to extract compressed N1B scenes. This software was used as a template to create a suite of C programmes to read the N1B files and apply at-sensor radiometric calibration to thermal channels. At the core of the pre-processing suite were two key modules. The scanline utility extracts the compressed 10-bit data to normal 16-bit floats and applies a big-endian to little-endian conversion, to create images in the PGM format. The radiance utility applies radiometric sensor calibration coefficients (as recorded by the sensor for each scene) to raw satellite DN values to create at-sensor radiance. Appendix A contains the source code for both of the programmes.

After continued discussion with DSS during the Autumn of 2009 the final product was provided with radiometric and geometric corrections applied by DSS in the GeoTiff image format.

3.2.4 Georeferencing

Georeferencing at DSS is undertaken in three stages (DSS, 2007), first an approximate location of a scene is derived using the time of satellite overpass and known orbit parameters to calculate Earth location. Then, Doppler cor-

rections recorded by DSS antenna controllers are used to correct for satellite clock drift to further improve positional accuracy. Lastly, the Landmark software package developed by DSS is used to adjust time, yaw, roll and pitch parameters of a scene until coastline features in a scene fit known locations of the true coastline (DSS, 2007). For this project the AVHRR georeferencing was optimised by DSS for the centre of London. Therefore only scenes which covered the whole of London, and not at the edge of the swath were processed. To correct for off-nadir sensor view angle a nearest-neighbour correction was used by DSS to average all pixels to 1.1Km resolution.

A set of four sample rectified scenes in the GeoTiff format was provided by DSS for quality control checking. It was noted that the images provided with geometric correction had apparently poor positional accuracy to surface features recorded at sensor, with errors of 1-2km visible. The Isle of Wight was used as a test site to evaluate the geometric correction. A 1:250,000 vector outline of the British Isles derived from Aerial photogrammetry was provided by Getmapping plc, to provide a standardised data-set that could be shared with DSS to improve georectification. Using this data it was possible to compare pixels in AVHRR band 2 with the Isle of White vector outline. Band 2 was chosen as it provides a high contrast between water (minimal reflectance in the near-infrared wavelengths) and the land surface. The sample AVHRR scenes appears to be incorrectly shifted in a South-Westerly direction by approximately 1.5km. It was discovered that this was due to a bug in the correction software developed by DSS. This was corrected and the scenes were re-processed with the correct geometric correction within the order of 1 pixel (<±1.1km).

3.2.5 Radiometric correction

The purpose of radiometric calibration is to correct images for differences in sun angle and sensor bias at the time of image capture, to derive top of atmosphere albedo (optical/IR bands) and brightness temperature (thermal bands) that are comparable between scenes throughout a time-series (Mather,

2004). The process of AVHRR scene radiometric calibration involves two stages. First, the raw sensor voltages are converted to at-sensor radiance, based on pre-launch sensor calibration (NOAA, 2009, 2012). An in-flight calibration of thermal bands to account for thermal sensor degradation during orbit is also applied at this stage (NOAA, 2009). Second, at-sensor values are corrected to represent albedo/brightness temperatures by accounting for surface reflectance, solar azimuth and sensor viewing angle (Cracknell, 1997; NOAA, 2012).

To convert digital numbers from the visible and near infrared bands (1, 2 and 3A, see Table 3.6) pre-launch calibration coefficients are used (Cracknell, 1997; NOAA, 2009). These coefficients represent the relationship between the output of the radiometer and known radiation values as measured in laboratory conditions prior to sensor launch (Cracknell, 1997). The calibration coefficients are represented in the form of a simple linear regression relationship between radiometer value and known radiation levels (NOAA, 2009). Albedo is then calculated by multiplying the AVHRR signal by the slope of the regression and adding the intercept value to the result, and multiplying this by the Earth-Sun distance to account for top of atmosphere incident solar radiation. The variation in solar elevation angle is accounted for by using the solar zenith angle and assuming Lambertian reflectance of the surface to derive top of atmosphere albedo. Whilst the Earth's surface rarely exhibits perfect Lambertian reflectance this correction may be approximate to the first order and suitable when processing a large archive of imagery (Dash et al., 2002; Mather, 2004),

A similar technique is used for the thermal channels (3B, 4 and 5). However the AVHRR sensor also performs in-flight calibration for the thermal radiometer sensors. During each scan-line the sensor views three objects; deep space, the Earth and an internal blackbody calibration target (Cracknell, 1997; NOAA, 2009). Given the known temperature of deep space and the temperature of the calibration target as measured by an on-board platinum resistance thermometer an adjusted linear regression equation using the in-orbit response of the sensor can be derived. This is then used to covert

digital numbers for each scan-line to at-sensor thermal radiance. Since the launch of NOAA-15 (13th May 1998) the level 1b data format incorporates non-linear radiance corrections for AVHRR thermal bands 4 and 5 which provides a more realistic representation of the sensors response over a range of radiance values (NOAA, 2009). The calibration coefficients from measurements of deep-space and the calibration target are stored at the beginning of each row of scan-line data. The process of extracting these for each scan-line is shown in "scanline.c" (Appendix A.1) and the implementation of these for nonlinear conversion of thermal radiometer digital numbers to at-sensor radiance is shown in "radiance.c" source code (Appendix A.2).

3.2.6 Data ordering

Once the file format and image georeferencing options were decided (see Section 3.2.4) an order was placed with DSS for the required time series of AVHRR data. The selection of scenes to be included in the archive was done through the DSS Pass Database which generates a HTML list of available scenes for a given location along with the scene attributes including percentage cloud cover. The Pass Database was queried to show all AVHRR scenes for the summer months (May to September) between 1985 and 2008 with less than fifty-percent cloud cover. The results from this query were then parsed into text format using the html2text utility and each scene was assigned a unique reference based on a concatenation of satellite number, date, time and orbit number.

The list was used to create a selection of scenes for ordering from DSS. To minimise atmospheric effects and data gaps from cloud coverage it was decided to limit the order to scenes with a zero-percent cloud cover. A selection of scenes from the text list where cloud coverage was zero was created and sent to DSS for processing. The completed order was delivered by DSS in January 2010 and comprised of 2,416 five-band GeoTiffs calibrated to top of atmosphere albedo and brightness temperature.

3.2.7 AVHRR meta-data database

To efficiently manage the archive of scenes an AVHRR meta-data database was created (Figure 3.4). The database was created with a table containing AVHRR scene meta-data for each scene, extracted from the DSS database during ordering (Section 3.2.6). This table used a compound primary key generated from the satellite number and overpass date, time and orbit number. A second table containing the scene primary key and the location of each of the scenes (including file name) on disk was also created. The two tables were linked by a one-to-one relationship based on the scene primary key, meaning that scenes stored on disk can be selected for processing based on their meta-data attributes (e.g. time of overpass) (Figure 3.4). Furthermore, the AVHRR meta-data database is stored alongside the MIDAS database meaning that it is possible to select measurements from the weather station data within the same hour as AVHRR scene satellite overpass for comparison. Figure 3.4 shows a schema diagram of the MIDAS and AVHRR database tables. The figure also includes an example of a relationship which can be derived between the AVHRR meta-data table and hourly air data table, in this case to select measurements from the hourly air data for the same time and date as an AVHRR overpass. This database structure formed the basis for the software framework discussed in Chapter 4, which enabled the spatio-temporal pairing of EST and air temperature measurements for analysis (Section 5.2.3).

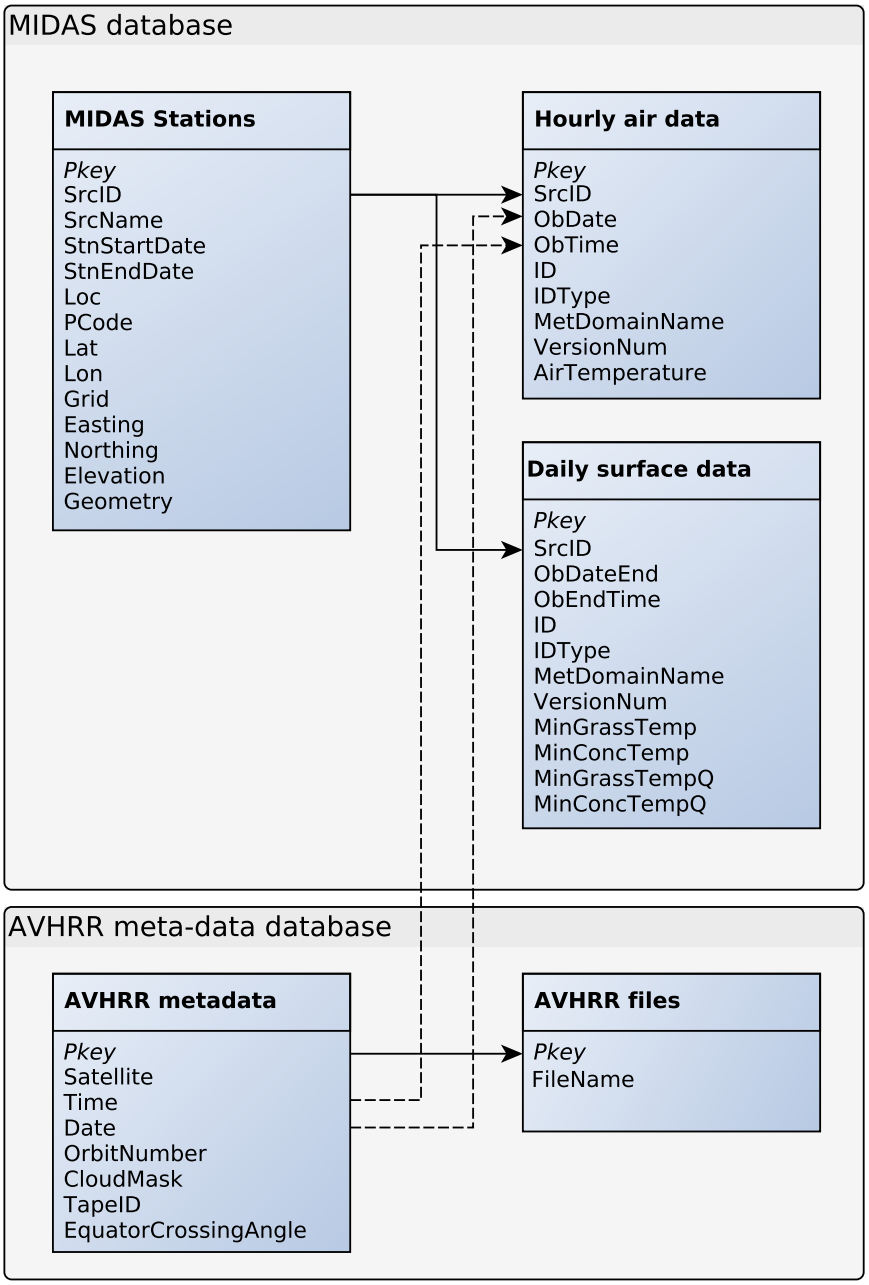


Figure 3.4: Entity-relationship model showing the MIDAS and AVHRR database tables and an example of a relationship between the two.

3.2.8 Cloud screening

During the data selection process, scenes with zero percent cloud cover were selected for ordering, but, the DSS cloud mask only covered the point in the centre of the Greater London Authority (GLA) administrative boundary, and not the wider London area. The rural station at High Wycombe outside of the

GLA boundary was used during the urban heat island intensity calculation (see Section 6.2.2) and therefore an additional test for cloud coverage not filtered by the DSS pass database was undertaken for the daytime scenes. To achieve this scenes were first masked so that pixel values were only present for the GLA and rural station locations, so as to prevent cloud contamination from regions of the scene outside the study from being detected. The spatial extents of the mask were derived in the Quantum GIS package using the GLA extents polygon and a 7x7 pixel buffer (59.29km^2) around the location of the rural station. Masked scenes which contained cloud cover were then identified using an empirical approach based on the difference between EST and diurnal minimum grass surface temperatures from the MIDAS database. This was achieved by comparing the minimum top of atmosphere (TOA) brightness temperature of each scene to its corresponding daily minimum grass surface temperature recorded by the London Heathrow weather station.

If any pixel in the AVHRR scene had a TOA more than 3°C lower than the daily minimum grass surface temperature then the scene was tagged as cloud contaminated. The 3°C threshold was selected on the basis that a ±3°C uncertainty may be expected to exist between uncorrected AVHRR TOA temperature and corresponding surface temperature (Cooper and Asrar, 1989). This resulted in 1,073 scenes from the total of 1,718 daytime scenes, identified as cloud free over the Greater London Authority and the rural weather station. Furthermore a further 68 scenes were identified as cloud free over the Greater London Authority area but not over the rural weather station. These scenes could be used in addition to the aforementioned 1,073 scenes when measurements from the rural station were not required (see Section 5.2.3).

3.2.9 Atmospheric correction of optical AVHRR data

Earth observed images in the optical wavelengths often need to be corrected for atmospheric absorption and scatter, which alter upwelling radiance between surface and satellite sensor (Teillet, 1992; Rahman and Dedieu, 1994; Kalluri and Dubayah, 1995; Lillesand et al., 2004; Nagol et al., 2009; Er-

bertseder et al., 1999). There are three principal methods for atmospheric correction of optical Earth observed data; image-based methods, the empirical line method and radiative transfer modelling (Mather, 2004).

Image-based methods work on the assumption that some land cover types such as water, or areas in deep shadow have close to zero reflectance in the near-infrared bands and so offset in these bands provides an estimate of atmospheric attenuation (Hadjimitsis et al., 2004; Lu and Weng, 2006). One implementation is the histogram method which works by creating a histogram of all pixel values in each of the bands and using the values for pixels containing non-reflecting land cover types (e.g. water or deep shadow) as a first order approximation of atmospheric attenuation (Mather, 2004). The difference between zero and the lowest value in the histogram for these areas is then subtracted from all the other pixel values in the band (Lu et al., 2002; Hadjimitsis et al., 2004).

An alternative to the histogram method is to use a regression method where pixel values for dark surface features in the near-infrared bands are plotted against pixel values from each of the other visible/near-infrared bands in-turn (Mather, 2004). A least-squares best-fit linear regression is then computed for each set of pixels (Mather, 2004). The offset on the x-axis between a near-infrared band and the band in question represents an estimate of atmospheric path radiance for each spectral band (Mather, 2004). This is then used as a correction for all the pixels in the specified band. Image-based methods for atmospheric correction are often favoured due to their simplicity and not requiring ancillary data (Lu et al., 2002; Hadjimitsis et al., 2004). However, such methods have been found to be highly subjective given the requirement for dark objects in the image (Franklin and Giles, 1995).

The empirical line method is similar to image-based methods except that it uses ground measurements of reflectance for known bright and dark targets captured using a radiometer at the same time as the satellite overpass. The values for dark and light objects measured at both the sensor and on the ground are then used to plot a line, the x-offset of which provides an esti-

mation of atmospheric radiance (Mather, 2004). The obvious disadvantage of this method is the requirement for synchronised ground measurements of surface reflectivity, along with the identification of suitable light and dark targets, meaning that this method is unsuitable for time-series analysis where ground measurements haven't been taken (Mather, 2004).

The last alternative is to use a physics-based radiative transfer model of atmospheric interactions to correct scenes (Hadjimitsis et al., 2004; Lu and Weng, 2006). Such models are normally parametrised either with empirical atmospheric profile data captured at the time of satellite overpass (e.g. from radiosonde or weather radar), or use a generalised atmospheric model for the area of interest (Mather, 2004). However, a number of studies have raised questions as to the utility of radiative transfer models (Lu et al., 2002; Hadjimitsis et al., 2004; Mather, 2004; Nagol et al., 2009). One of the problems is that radiative transfer models are computationally expensive, making them unsuitable for processing a large time-series of scenes (Lu et al., 2002). Furthermore, the use of generalised atmospheric profiles when local empirical atmospheric data is not available to parametrise the models reduces their accuracy, and the extent of the loss of accuracy is not quantifiable (Mather, 2004).

Nagol et al. (2009) used AVHRR data to compare uncorrected top of atmosphere NDVI with NDVI from the Pathfinder AVHRR Land (PAL) data-set, which uses a radiative transfer equation to correct for Rayleigh scattering and ozone absorption. They showed that during clear sky conditions (aerosol optical thickness (AOT) < 0.05) PAL improved NDVI uncertainty by 0.0377 and during average conditions (AOT = 0.05 $\geq$ 0.25) NDVI uncertainty was improved by 0.0765 compared to uncorrected top of atmosphere NDVI. As a result, Nagol et al. (2009) stated that radiative transfer modelling techniques may only lead to a minor improvement in data quality during clear-sky conditions.

In consideration of the long-time series of AVHRR data being employed in this study, the summary of the literature indicates that image based methods

for atmospheric correction are more suited to processing of individual scenes (Mather, 2004), and can be highly subjective and inaccurate (Franklin and Giles, 1995; Lu et al., 2002). Furthermore, the aforementioned results from Nagol et al. (2009), and the computational requirements of radiative transfer models (Lu et al., 2002) indicate that such methods are also not operationally valid for long time-series Earth observed data. Therefore, given the lack of consensus in the literature as to a standard operational approach to perform atmospheric correction of historical data, in the absence of in-situ atmospheric profile data (Teillet, 1992; Erbertseder et al., 1999), the decision was made to utilise the AVHRR optical bands without atmospheric correction. An estimate of error propagation as a result of atmospheric attenuation in the optical bands is included in Section 3.2.12.

3.2.10 Atmospheric correction of thermal Earth observed data

Similar to optical bands, atmospheric attenuation may increase or decrease the upwelling thermal radiation from the Earth's surface measured by the satellite sensor (Kalluri and Dubayah, 1995; Campbell, 1996; Lillesand et al., 2004). Atmospheric absorption of thermal radiation is minimised in the 3-5μm and 8-14μm wavelengths of the electromagnetic spectrum (Lillesand et al., 2004), within which most thermal scanners operate (Lillesand et al., 2004).

Within the 10 to 12μm wavelength range of the electromagnetic spectrum, water vapour, carbon dioxide (CO_2), nitrogen (N_2) and ozone (O_3) have the greatest effect upon atmospheric thermal radiation attenuation (Price, 1987). During clear sky and stable weather conditions, the influence of aerosols is generally a magnitude of order smaller than molecular effects except when the atmosphere is hazy (Price, 1987). Compared to water vapour, many studies regard the effects of CO_2, N_2 and O_3 as negligible (Price, 1987; Kalluri and Dubayah, 1995). Furthermore, as CO_2 and N_2 mixing levels are almost constant, their effects can often be easily accounted for (Kalluri and Dubayah, 1995). As such, attenuation of the thermal radiation by water vapour is of key

importance (Price, 1987; Kalluri and Dubayah, 1995).

Atmospheric correction can be achieved using either single channel or split-window techniques (Erbertseder et al., 1999; Yu et al., 2008). For sensors capturing thermal radiation in a single channel, a radiative transfer model can be used to correct for atmospheric attenuation in the thermal wavelengths (Price, 1987; Cooper and Asrar, 1989; Goetz et al., 1995; Kalluri and Dubayah, 1995; Schmugge and Schmidt, 1998; Dash et al., 2002; Jimenez-Munoz and Sobrino, 2008; Sobrino et al., 2008; Yu et al., 2008). Radiative transfer models can be parametrised by either empirical atmospheric profile data or generalised atmospheric models (Price, 1987; Cooper and Asrar, 1989).

Sensors which capture thermal data in two or more different, but spectrally close bands (e.g. 11μm and 12μm) can utilise split-window techniques (SWT). SWT corrections are based on the difference in atmospheric absorption of radiation between bands of different wavelengths, and produce one EST value from the linear combination of brightness temperatures from two or more bands (Becker and Li, 1990b; Yu et al., 2008).

Kalluri and Dubayah (1995) evaluated five techniques for atmospheric correction of AVHRR thermal data during the First International Satellite Land Surface Climatology Project (ISLSCP) Field Experiment (FIFE). The study compared AVHRR band 4 and band 5 values from three images and ground infrared thermometer measurements taken over three days. The correction procedures included three methods based on radiative transfer models (RTM) and two split window approaches.

Table 3.7 lists the five methods and their average error over the three dates examined. A ±3°C threshold between ground measurements and EST was selected to evaluate the success of correction models, based on the minimum precision of which EST can be measured from AVHRR data as derived by Cooper and Asrar (1989). Coincident radiosonde data were used to parametrise the LOWTRAN model and to derive localised coefficients for the SWTs.

SWT coefficients are normally based on generalised atmospheric profiles, but

in this case Kalluri and Dubayah (1995) wanted to evaluate whether localised coefficients would improve the SWT corrections. Kalluri and Dubayah (1995) found that all three algorithms that did not use the radiosonde data failed the 3°C criteria on all three days. The study also found that the uncorrected satellite measurements for band 4 were on average, over the three days, 1.16°C closer to the ground truth than the best correction method (Table 3.7). Kalluri and Dubayah (1995) state their concern at this as it indicates that atmospheric corrections introduced additional error. However, the study does note that emissivity and off-nadir viewing angles were not corrected for in the evaluation which would improve the corrections (Kalluri and Dubayah, 1995).

Method	Type	Radiosonde (yes/no)	ΔT (min)	ΔT (max)	ΔT ($\overline{x}$)
Uncorrected channel 4	-	No	-3.36	-0.95	-1.84
LOWTRAN linear correction	RTM (linear equation)	Yes	2.24	4.72	3.19
Localised split window	SWT (with localised coefficients)	Yes	2.19	4.03	3.0
McClain et al. (1983)	SWT (with non-localised empirical coefficients)	No	4.14	4.8	4.21
Price 1984	RTM (linear equation)	No	4.48	5.23	4.77
Becker and Li (1990b)	SWT (derived from least squares fit of LOWTRAN6 simulated data)	No	3.14	4.17	3.67

Table 3.7: Atmospheric correction techniques and their average errors as evaluated by Kalluri and Dubayah (1995). *Note:* $\Delta T = T_{measured}$- *EST.*

A previous study by Cooper and Asrar (1989) used four AVHRR scenes from four days to evaluate six models for atmospheric correction (two RTM and four SWT), including the SWT by McClain et al. (1983). In contrast to Kalluri

and Dubayah (1995), Cooper and Asrar (1989) found that only the SWT correction by McClain et al. (1983) produced ESTs within ±3°C over the four days. Kalluri and Dubayah (1995) note this discrepancy in their study and they suggest that it is the result of differences in the atmosphere conditions between the two studies, although further work is needed to qualify this.

Despite this apparent ambiguity between studies, later research by Erbertseder et al. (1999) and Yu et al. (2008) promotes the application of SWTs due to their robustness and simplicity over traditional methods (Erbertseder et al., 1999). Erbertseder et al. (1999) and Yu et al. (2008) also cite the lack of consensus in the literature over which RTM method provides the best results. In their evaluation of nine SWTs Yu et al. (2008) found that older algorithms, such as those by Price (1984) and Becker and Li (1990b), performed on-par with newer algorithms such as that developed by Caselles et al. (1997), and hypothesised that the greatest variations in ESTs between methods were caused not by atmospheric attenuation but by emissivity based errors (Yu et al., 2008). As a result of the findings from the literature to correct for atmospheric attenuation in the thermal bands over the long time-series of AVHRR imagery employed in this study, a split-window correction was applied. The method proposed by Becker and Li (1990b) was used as it has been implemented operationally by the German Aerospace Centre in conjunction with an emissivity correction (see Section 3.2.11). The implementation of the SWT is described in Section 3.2.12.

3.2.11 Emissivity correction of thermal Earth observed data

Emitted electro-magnetic radiation is a function of an object's kinetic temperature (Lillesand et al., 2004). Planck's law describes the spectral exitance from a blackbody as a function of its temperature (Campbell, 1996). However, this law only holds for hypothetical objects which are perfect absorbers and emitters of thermal radiation (i.e. a blackbody) (Campbell, 1996; Lillesand et al., 2004). Real-world objects rarely exhibit blackbody properties, and have

variable absorption and emittance characteristics depending on the material in question (Lillesand et al., 2004). Emissivity is a ratio which describes how effectively an object radiates thermal energy compared to a blackbody at the same temperature (Lillesand et al., 2004). Therefore, an emissivity correction must be applied to thermal radiation measured by a satellite sensor before deriving an estimate of surface temperature using the radiant flux density-wavelength blackbody relationship defined by Planck's law (Campbell, 1996).

There are a range of methods for estimating emissivity using ancillary data and Earth observed data in the thermal and optical bands (Dash et al., 2002; Sobrino et al., 2008). Table 3.8 lists the current methods available as defined by Dash et al. (2002) and Sobrino et al. (2008). Two types of emissivity estimation methods exist, those deriving relative emissivity independent of surface temperature and those deriving absolute emissivity used to correct remotely sensed surface temperature estimates (Sobrino et al., 2008). For thermal Earth observation, absolute emissivities are required to derive EST (Sobrino et al., 2008), and as such will be the focus of discussion here.

Table 3.8 shows that there are five methods for calculating absolute emissivity which can be used to derive EST from Earth observed thermal radiance. Two of these methods require emissivity information (EBM and ICL) and so cannot be used when emissivity is not known in advance. From Table 3.8 it can be seen that there are three methods which can be used to derive EST without emissivity information.

The TES algorithm provides absolute emissivity, without requiring emissivity values (Sobrino et al., 2008). Developed for the ASTER sensor by Gillespie et al. (1998), the TES algorithm uses hyper-spectral thermal data from the ASTER sensor, along with pre-defined empirical relationships between spectral contrast and minimum emissivity to invert the Plank equation for measured thermal radiance (Gillespie et al., 1998). The algorithm is capable of recovering both emissivity and EST from ASTER scenes with accuracies of ±0.015 and ±1.5K respectively. However, the requirement for hyper-spectral thermal data makes TES unsuitable for long time-series of scenes acquired

Method name	References	Absolute / Relative emissivity	Requires emissivity data	Description
Normalised Emissivity method (NEM)	Gillespie (1985)	Relative	No	An initial approximation of maximum surface temperature is used to estimate emissivity for all bands.
Thermal Infrared spectral indices (TISI)	Becker and Li (1990a)	Relative	No	Construct indices's of emissivity independent of temperature to derive emissivity
Alpha derived emissivity (ADE)	Kealy and Gabell (1990)	Relative	No	Use the natural log on Wien's approximation of Plank's law to eliminate surface temperature and derive emissivity
Emissivity bounds method (EBM)	Jaggi et al. (1992)	Absolute	Yes	Using estimates of emissivity bounds, a temperature estimate is made, which is then refined to estimate a tighter bound on emissivity
Ratio algorithm (RA)	Watson (1992)	Relative	No	Derive ratio of spectral emissivity using an approximate estimate of temperature
Normalised Difference Vegetation Index (NDVI) methods	Van De Griend and Owe (1993); Valor and Caselles (1996)	Absolute	No	Empirically derived relationship between NDVI and surface emissivity
Grey body Emissivity Method (GEM)	Barducci and Pippi (1996)	Absolute	No	For grey bodies such as soil, applies assumption that emissivity has a flat spectrum for wavelengths greater than 10μm to estimate average emissivity.
Day/night methods (DNM)	Goïta and Royer (1997)	Relative	No	Twice daily co-registered day/night images using difference between surface temperature to estimate emissivity using the TISI method
Temperature Emissivity Separation (TES) algorithm	Gillespie et al. (1998, 1999)	Absolute	No	Compensates for reflected downwelling irradiance and uses pre-defined empirical relationships to estimates emissivity from multi-spectral thermal data
Image classification (ICL)	Snyder et al. (1998)	Absolute	Yes	BRDF models based on laboratory specta measurements of different land surface materials

Table 3.8: Emissivity correction methods reviewed by Dash et al. (2002) and Sobrino et al. (2008).

by older sensors with fewer thermal bands such as the AVHRR (Dash et al., 2002).

The Grey body Emissivity Method (GEM) techniques can be used for non-hyperspectral data (Barducci and Pippi, 1996). GEM works by assuming that emissivity is a slow varying function of wavelength and has a flat spectrum for wavelengths greater than 10μm, meaning that the slope of the emissivity spectrum is negligible (Barducci and Pippi, 1996). Using these assumptions Barducci and Pippi (1996) derived a formula for calculating grey-body emissivity. The model uses an iterative maximum likelihood approach, with an input trial temperature from which an estimate of emissivity is made. The emissivity estimate is then constrained within reasonable bounds and used to estimate a new temperature value which in-turn is used to estimate a new emissivity value. This process is repeated until model convergence (emissivity and temperature are within given bounds), and was shown to be capable of estimating temperature within 0.001K (Barducci and Pippi, 1996). The disadvantage of this method is that the assumptions for grey-body behaviour are not normally valid for the urban surface except for bare soil and other natural materials (Sobrino et al., 2008).

A number of methods for estimating emissivity using the normalised difference vegetation index (NDVI) derived from images captured in the optical bands at the same time as thermal images have been developed (Van De Griend and Owe, 1993; Valor and Caselles, 1996; Sobrino et al., 2008). Van De Griend and Owe (1993) used fifty-nine samples of emissivity and NDVI in the Botswana savanna to derive a logarithmic relationship between the two. The study showed an r^2 correlation value of 0.89 at the 0.01 confidence level, covering a range of land cover types including sand, bare soil and dense vegetation. Van De Griend and Owe (1993) state that thermal reflectance from vegetation takes place in the leaf surface layer, and is therefore indirectly related to pigment absorption in the red portion of the electro-magnetic spectrum, captured by NDVI. The derived relationship shows that the higher the NDVI, the greater the thermal emissivity. The disadvantage of this method is that the empirically derived relationship doesn't necessarily hold for dif-

ferent land-cover environments (Dash et al., 2002). However it has been adopted for other areas including Europe (Valor and Caselles, 1996; Dash et al., 2002; Tungalagsaikhan and Guenther, 2007) and South America (Valor and Caselles, 1996).

3.2.12 Atmospheric and emissivity corrections of AVHRR data

In order to perform atmospheric and emissivity corrections on long time-series AVHRR data, the chosen methods must be able to perform a valid correction without a large computational expense to minimise processing time of the large number of scenes. Furthermore, the method cannot rely on ancillary data as this is often not available over a long time-series (Teillet, 1992; Erbertseder et al., 1999).

The German Aerospace Centre (DLR) provide an operational approach which derives estimated surface temperatures using a combined split-window technique for correcting atmospheric attenuation (Becker and Li, 1990b) and a surface emissivity correction based on the use of the Normalised Difference Vegetation Index (NDVI) using AVHRR bands 1 and 2 (Van De Griend and Owe, 1993). This approach has been found to be computationally efficient when processing large volumes of scenes in an automated manner (Tungalagsaikhan and Guenther, 2007). Given the volume of data and the requirement for an emissivity correction procedure applicable for time-series data without a priori information it was decided to employ this approach to derive EST from the AVHRR thermal bands four and five. In this approach, estimated surface temperature (EST) was derived on the basis of bands 4 and 5 of the AVHRR instrument (Equation: 3.1):

$$\begin{aligned} T =& 1.274 + \frac{T_4 + T_5}{2} * (1 + 0.15616(\frac{1-e}{e} - \frac{0.482de}{e^2})) \\ &+ (\frac{T_4 - T_5}{2 * (6.26 + 3.989(\frac{1-e}{e}) + \frac{38.33de}{e^2})}) \end{aligned} \tag{3.1}$$

Where T is the estimated surface temperature (K), T_4 and T_5 are the black-body temperature of bands 4 and 5 of the AVHRR instrument (top of atmosphere brightness temperatures) and e and de represent the emissivity correction factors given by:

$$e = \frac{(e_4 + e_5)}{2} \tag{3.2}$$

$$de = e_4 - e_5 \tag{3.3}$$

where e_4 and e_5 represent the emissivity of bands 4 and 5 of the AVHRR instrument and are estimated on the basis of:

$$e_4 = 1.0094 + 0.047ln(NDVI) \tag{3.4}$$

$$e_5 = e_4 + 0.01 \tag{3.5}$$

In Equation 3.4 the NDVI (Normalised Difference Vegetation Index) is scaled linearly between 0-255 (Van De Griend and Owe, 1993):

$$NDVI = ((\frac{B_2 - B_1}{B_2 + B_1}) + 1) * 127 \tag{3.6}$$

In the emissivity correction procedure (Equation 3.1), no atmospheric correction for bands 1 and 2 was undertaken in relation to the calculation of NDVI (Equation 3.6). Given the number of scenes under investigation and the fact that there is a lack of consensus in the literature as to a standard operational approach to atmospheric correction of historical data in the absence of in-situ atmospheric profile data (Teillet, 1992; Erbertseder et al., 1999; Hadjimitsis et al., 2004) the decision not to perform atmospheric correction to the optical bands for NDVI calculation could be justified.

Nonetheless, the lack of atmospheric correction potentially introduces an uncertainty in the NDVI calculations ranging from ±0.09 during clear conditions (aerosol optical thickness (AOT) < 0.05) to ±0.13 during average conditions (AOT = 0.05 $\geq$ 0.25) (Nagol et al., 2009). This may result in error propagating in to the emissivity correction. In order to assess this, one can invert Equation 3.4 to obtain expected NDVI for a typical emissivity of an urban surface ($\varepsilon \leq 0.92$; (Nichol, 1994)) and vegetation ($\varepsilon \geq 0.98$; Dash (2005)). By adding the expected NDVI error of ±0.13 for average sky conditions to these and applying Equation 3.4 one obtains an emissivity error of ±0.022 or greater for urban surfaces, and ±0.007 for vegetation. In turn, this emissivity error will propagate into the subsequent estimation of surface temperature. On the basis of work done by Sobrino et al. (1991) which found that emissivity accuracy needs to be within ±0.005 to get estimated surface temperature below ±0.4K, and the work of Schädlich et al. (2001) and Dash et al. (2002) who found for mid-latitude areas an emissivity error of ±0.025 gave an error of ±2K in estimated surface temperature, then an initial NDVI error of ±0.13 may be expected to result in ~±2K error over urban pixels and ~±0.4K over vegetated areas.

To facilitate pre-processing and analysis of the large number of AVHRR images a Python suite of raster processing functions "PyRaster" was written (see Section 4.2). This includes a Python AVHRR (PyAVHRR, Section 4.2.2) module with the `lst_DLR` function to apply the above atmospheric and emissivity correction to the AVHRR data (see Listing 4.6, Section 4.2.2). The PyAVHRR module was used to batch process the AVHRR daytime dual thermal band scenes to EST. Of the potential 1,141 cloud free daytime scenes (1,073 over the GLA and rural station), 851 were dual thermal channel and so could be processed to EST (803 covering GLA and rural station). In this manner two EST data-sets were created, termed processing 'level 2' to distinguish them from the initial data provided by DSS. Data-set 'a' contained the 851 EST scenes for the GLA and data-set 'b' contained the 803 EST scenes for the GLA and the rural station (Figure 3.3). The creation of two data-sets in this way ensured that for analysis which focused on the GLA and didn't require data

for the rural weather station location (see Chapter 5) the maximum number of EST measurements were available. Two additional tables were created in the AVHRR meta-data database containing the filenames of the new EST scenes and the primary keys of the originating scene (the same as the 'AVHRR files' table in Figure 3.4). As such, derived scenes of EST could be linked to the meta-data attributes in the 'AVHRR metadata' table and so could be selected by date and time.

3.3 Summary

This chapter has described the pre-processing methodologies performed to create a spatial database containing the locations of UKMO weather stations, and their records of hourly air data and daily surface data from the MIDAS data-set. The chapter has explored the rigorous data-filtering strategy employed in order to produce a robust time-series of error free temperature measurements from the MIDAS data. Furthermore, an operational methodology to process the long time-series of AVHRR thermal band data (1985-2008) to EST is presented, based on an appraisal of existing methods from the literature. In association with the AVHRR processing, the creation of the AVHRR meta-data database allows AVHRR scenes to be selected based on the attributes of each satellite overpass (Figure 3.4). As discussed in Section 4.3 this means that scenes can be selected for the same time and date as MIDAS temperature measurements, enabling integrated analysis of the two (see Section 5.2.3). The MIDAS and AVHRR data-sets created in this chapter form the basis for the analysis carried out in the remainder of the project. The development of a software framework and uniform data model for integration of the two data is discussed in Chapter 4.

Chapter 4

A software framework for integrated processing and analysis of thermal Earth observed data and weather station temperature measurements

4.1 Design and implementation

Despite the potential for time-series satellite Earth observed data to provide invaluable information about the Earth's surface (Latifovic et al., 2005), the utilisation of such data is often hindered by operational difficulties such as large data volumes and complicated processing methodologies (Latifovic et al., 2005). Furthermore, existing software packages only provide limited tools for the analysis of time-series spatial data-sets, and the development of such tools is still an area of active research (Worboys and Duckham, 2004). Therefore, in order to perform processing and analysis of the AVHRR and MIDAS tem-

perature data (Sections 3.1 and 3.2) this project developed a software framework building on the spatial databases created during pre-processing (Section 3.2.7). The aim of the framework is to enable the development of analytical routines using the AVHRR and MIDAS data in an integrated and computationally efficient manner.

4.1.1 Identifying framework requirements

Prior to software development it was important to identify the requirements of the framework to ensure that the aforementioned aim was met (Stephens et al., 2012). The identified framework requirements can be summarised as follows:

1. Data interface provision: The framework must provide an in-memory representation of AVHRR and MIDAS data structures, which is efficient at performing numerical analysis over individual elements, iteratively over the time-series. This will provide support for analytical scripts to be developed for time-series AVHRR and MIDAS data.

2. Spatial database support: The framework must be capable of reading and writing PostGIS spatial database tables and associated spatial meta-data for both MIDAS and AVHRR data. This will allow the framework to leverage the efficient computation and data-storage functionality of the PostGIS database for data analysis.

3. Integrated data management: The data interface must be capable of integrating AVHRR and MIDAS data using spatial and temporal meta-data so that analysis of EST and air temperatures for the same point in time can be undertaken.

4. Raster data support: The framework must be capable of reading and writing binary image files and associated spatial meta-data. This is because AVHRR files are stored as GeoTiff images and therefore read and write support of raster data formats is essential.

5. Rapid development support: The framework must support rapid prototyping of data analysis routines for exploratory data analysis. This will enable analytical and statistical routines to be developed in a timely fashion for analysis of the data to meet the project's aim and objectives.

6. Extensible: The framework should be able to leverage other software to perform high level analysis routines. This will enable the software framework to perform statistical testing of the data for analysis.

Over the last decade the Python programming language as been used extensively for computer-driven scientific research (Millman and Aivazis, 2011; Pérez et al., 2011; Terrel, 2011; van der Walt et al., 2011). Python is an open source, cross-platform, high-level interpreted language (Martelli, 2006), attributes which, along with the multitude of free and open source libraries available for scientific research make Python a popular choice for development in research projects (Martelli, 2006; Millman and Aivazis, 2011).

Python was chosen as the development environment for the framework as it could be used to develop tools which meet the identified requirements. Python was chosen for its command line interface which can be used for testing and exploratory analysis, and third-party libraries such as Numerical Python (NumPy) and Scientific Python (SciPy) which provide modules that can be used for scientific analysis of data (Table 4.1). The database infrastructure created during pre-processing (Section 3.2.7) formed the foundations of the framework, facilitating management and access of both MIDAS and AVHRR data-sets. A number of Python modules were created to interface with the database and AVHRR files, and provide a uniform data model for the development of analytical processes.

4.1.2 Creating a uniform data model

To meet the identified requirements it was necessary to design and construct a framework which had a centralised data model (a structure to contain and manipulate data) that could be used to represent both the AVHRR raster im-

agery and the MIDAS data. A uniform data model would achieve two things:

1. Analysis routines could be written once and used on both AVHRR and MIDAS data. This would mean that the same analysis could be performed on both AVHRR and MIDAS data using one piece of software, reducing the need to create data-specific processing routines.

2. Both data could be used in an integrated manner, with interchangeable input and output of data from either raster scenes or database tables. This is essential to be able to bring both EST and air temperature together to examine the relationship between them.

Numerical Python (NumPy) N-dimensional arrays are used extensively for scientific research in the Python language (Millman and Aivazis, 2011; van der Walt et al., 2011), and form the foundations of the Scientific Python software package (SciPy) (Jones et al., 2001). The NumPy array is an ideal container for raster data because it is memory and computationally efficient (van der Walt et al., 2011) and can use three dimensional arrays to represent multiband images. NumPy implements a strided data model for its arrays, meaning that an array can be stored once in memory but accessed from a number of different views simultaneously without duplication of the data (van der Walt et al., 2011).

Furthermore, the array data model incorporates methods for optimised vectorised operations (e.g. to perform arithmetic on each array element) (van der Walt et al., 2011) with fast pre-constructed iteration methods, making array operations extremely efficient (Oliphant, 2007; van der Walt et al., 2011). This style of data representation and manipulation is similar to the proprietary Interactive Data Language (Pérez et al., 2011), used by remote sensing packages such as the ENVI suite (ENVI, 2012). As such NumPy arrays have been used as the basis to develop a number of non-geospatial image processing libraries (Goncalves Silva et al., 2001; Lejdfors and Ohlsson, 2005; Oliphant, 2007).

NumPy also supports complex arrays with columns of different data-types. The structured array extends the standard NumPy array type to allow for

the storage of compound elements where columns of data may have different data-types (van der Walt et al., 2011). For example the first column of an array could be a date type, and the second a float. These advanced data-types allow NumPy arrays to represent complex data structures such as those found in spreadsheets and relational database tables (McKinney, 2011). The pandas library (Python data analysis library) builds on the NumPy record array structure to provide a library for multi-dimensional data management akin to database tables within the Python environment and is used in the framework to support operations on table data (see Table 4.1). Furthermore, the base class for the record array type is the NumPy array, meaning that it is possible to convert between standard NumPy array and record arrays using NumPy reshaping functions (Jones et al., 2001).

Given the aforementioned memory and computation efficiencies, and the ability to represent both homogeneous and complex data, NumPy arrays (N-dimensional and structured) were used to build a uniform data model in the framework to represent AVHRR image and MIDAS weather station data. Figure 4.1 shows the role of NumPy arrays as a uniform data model in the framework, In Figure 4.1 data from both AVHRR scenes and MIDAS database tables is loaded into memory as NumPy arrays (1). AVHRR scenes are first selected using the AVHRR meta-data database tables (e.g. select scenes by date) before loading, whilst MIDAS data can be selected and loaded directly from the database. The analysis then takes place using the NumPy array operators, integrating MIDAS and AVHRR data together if required (2). The results of the analysis processing can then be returned as either new raster files or database tables (3).

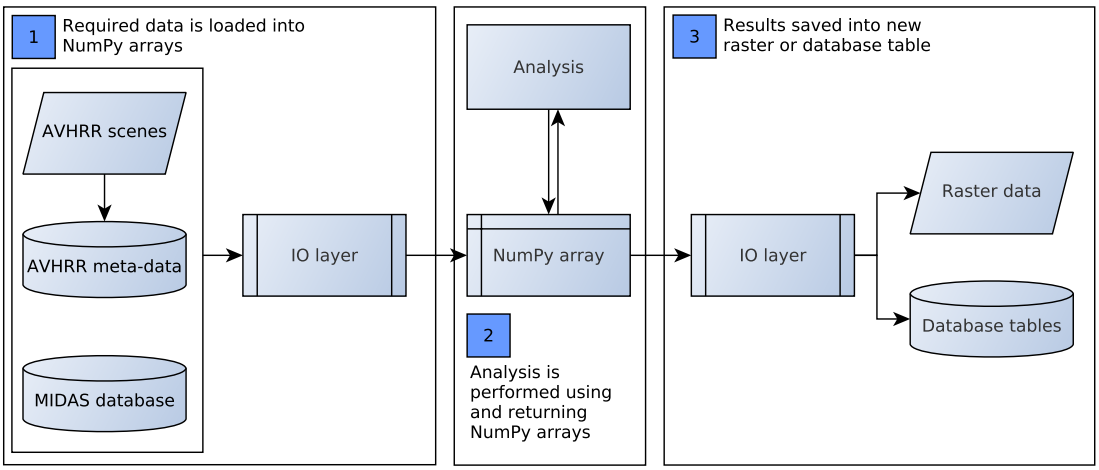


Figure 4.1: The role of NumPy arrays in the framework processing flow-line.

4.1.3 Framework architecture

The framework was designed around the processing flow-line described in Figure 4.1 and consists of a suite of Python modules stacked on-top of the PostGIS database and AVHRR file structure. Figure 4.2 shows that within the framework there are three layers, each of which comprises separate modules to perform specific functions. The connections between the different layers seen in Figure 4.2 represent flows of data and commands passed between the layers. The data layer is the foundation of the framework and consists of the MIDAS spatial database and the AVHRR files (Figure 4.2, see Section 3.2.7). The interface layer performs the conversion to/from NumPy arrays, and presents the data in the uniform data model for analysis (Figure 4.2). The analysis layer is at the top of the framework and represents the modules developed in this project to perform data analysis. A number of additional external modules were also used in the framework to help meet the identified objectives stated in Section 4.1.1, these are listed in Table 4.1.

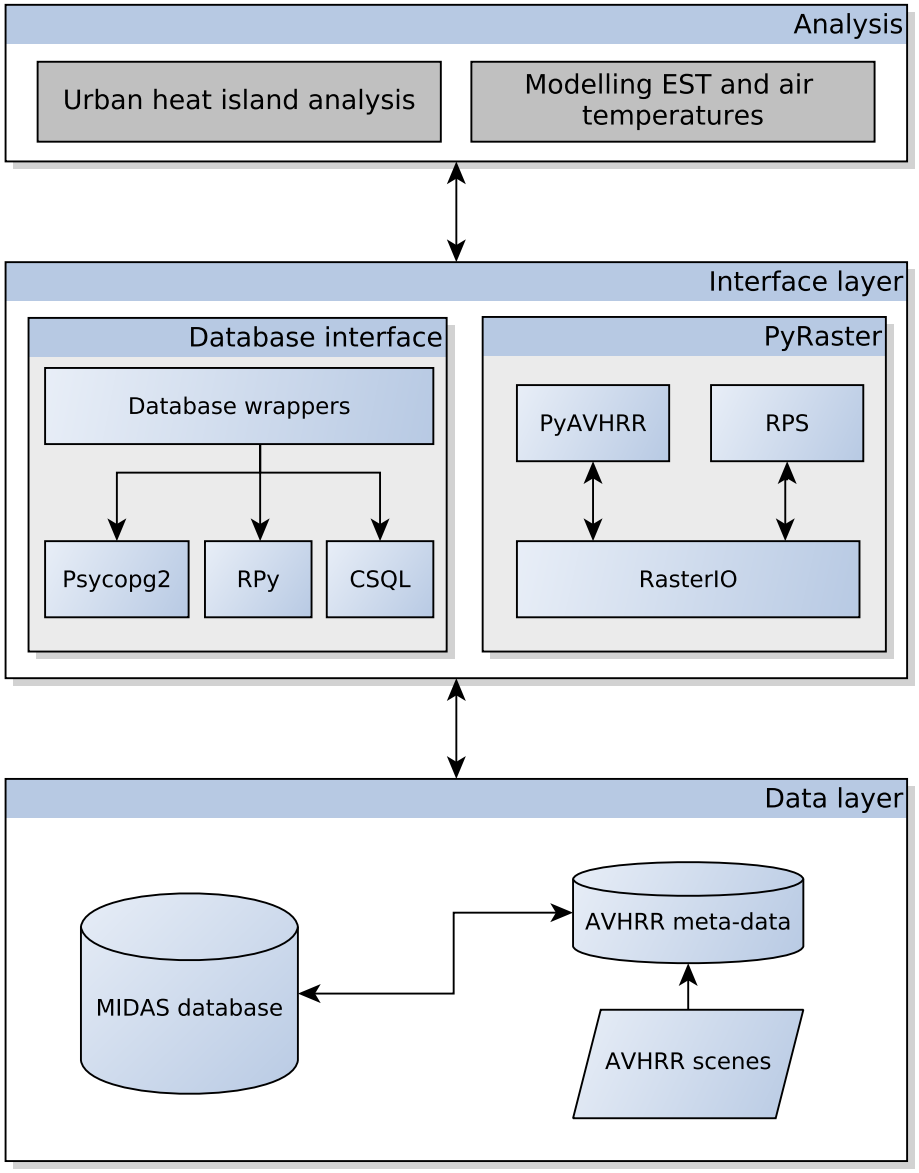


Figure 4.2: The architecture of the software framework showing the different layers and their modules.

Name	Description	Framework use	Reference
NumPy	Array structures and accompanying mathematical functions	In-memory representation of AVHRR and MIDAS data	Ascher et al. (1999)
GDAL	An abstract data model for geospatial data input and output	Read and write AVHRR scenes to NumPy arrays	GDAL (2012)
Psycopg2	PostgreSQL (and PostGIS) adaptor for Python	Read and write MIDAS data from PostGIS database to NumPy arrays	Varrazzo (2012)
RPy	A low-level interface to the R statistical language from Python	Perform statistical analysis on NumPy arrays	Gautier (2011)
SciPy	Scientific Python packages	Perform data analysis and statistical testing on NumPy arrays	Jones et al. (2001)
PyQt	Python bindings for the Qt GUI toolkit	Develop a graphical front end to the rasterIO module for the Quantum GIS package	Riverbank Computing Limited (2012)
pandas	Python library for advanced multi-dimensional (*pan*el *da*ta) management and analysis	Used to handle complex relational database tables when integrating MIDAS and AVHRR data	McKinney (2011)

Table 4.1: External Python libraries used in the software framework.

The interface layer (Figure 4.2) contains two groups of modules for accessing PostGIS database tables and AVHRR data. Both groups of modules support the NumPy array format. The PyRaster group contains modules for converting between AVHRR raster images and NumPy arrays as well as preprocessing functions for AVHRR data, and are discussed in Section 4.2.

The database interface contains a suite of modules for accessing PostgreSQL/-PostGIS databases, which are discussed in Section 4.3 below. Figure 4.2 shows the analysis layer built on top of the framework to undertake analysis of

London's spatio-temporal temperature dynamics, development of the analysis modules is discussed in Chapters 5 and 6.

4.2 PyRaster - Python spatial image processing

The PyRaster module group contains modules for interfacing with AVHRR raster images. The RasterIO module uses the Geospatial Data Abstraction Library (GDAL) (Table 4.1) to read and write raster images to and from NumPy arrays. Unlike the database connection modules used in the database layer, the GDAL bindings only provide a limited set of functionality (e.g. conversion of binary data formats and geospatial meta-data handling) and therefore the RasterIO module represents the largest section of development within the framework. As such the RasterIO module contains functions to read and write raster data and geospatial meta-data attributes, and convert these to Python data formats (e.g. NumPy arrays). The development of the RasterIO module is described further in Section 4.2.1. To perform AVHRR processing functions an additional Python module ('PyAVHRR') was created on-top of the RasterIO module, to perform sensor specific functions such as calculated estimated surface temperature (EST) and the normalised difference vegetation index (NDVI). Additionally, the RasterIO module was extended to form a raster calculator tool in the Quantum GIS (QGIS) package to aid exploratory analysis and development of processing the AVHRR data.

4.2.1 The RasterIO module

The RasterIO module performs the low-level data read/write functions to convert raster files on disk to and from NumPy arrays. The complete source code and documentation for the module is presented in Appendix B, Section B.1. To read a file from disk and covert it to a NumPy array three RasterIO functions are required; `opengdalraster`, `readrasterband` and

`readrastermeta` (Figure 4.3).

The `opengdalraster` takes the file name of a geospatial raster file and returns a GDAL pointer to the file. The `readrasterband` function reads a GDAL file pointer, the band number of the image to be read, a 'no data' value if the image contains missing values, and a True/False indicator of whether masking 'no data' values should be applied. The function then reads the data from the file specified by the pointer and returns a two-dimensional NumPy array containing pixel values of the input raster image (Figure 4.3). Listing 4.1 shows the code required to perform this operation (for brevity only the relevant lines are shown, see Appendix B, Section B.1 for the full source code).

Listing 4.1 shows that `readrasterband` first creates an empty two-dimensional NumPy data array, equivalent in size to the number of rows and columns in the input scene (Listing 4.1, Line 97). This array will hold the pixel values read from the raster file, the dimensions of which are defined by the GDAL pointer YSize and XSize methods. Next, a "for" loop is created to iterate over each of the rows of data in the image (YSize) which are accessed via the GDAL data pointer (variable 'band' in Listing 4.1). Within the loop three operations occur; first a binary representation of the pixel values in the row is read from the image using the GDAL read raster method on the GDAL pointer (Listing 4.1, Line 101). Second, the row is unpacked using the Python struct library, to extract the binary pixel values in the row into a tuple of numbers (Listing 4.1, Line 105). The data type for the values in the tuple is determined by global data type dictionaries defined in the RasterIO module which provide a mapping from GDAL (C) data types to equivalent Python data-types (see Appendix B, Section B.1). Once the row is unpacked the final operation in the loop is to add the row of pixel values to the empty array, replacing the zero values (Listing 4.1, Line 108). This process is then repeated for all the rows in the file until the array is populated and returned.

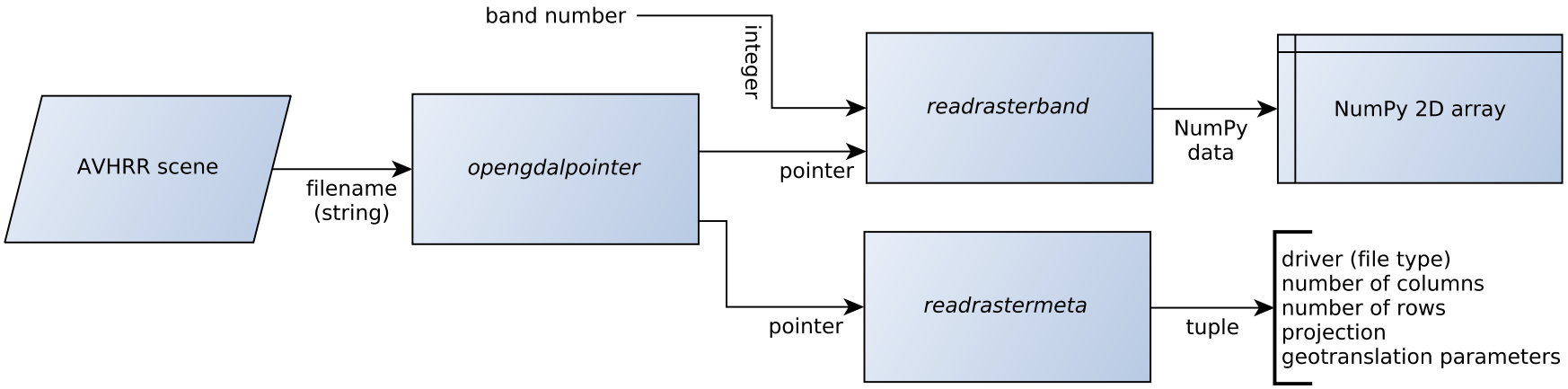


Figure 4.3: Data flow-line showing the RasterIO functions to read an AVHRR scene.

```
# create blank array (full of 0's) to hold extracted data [note Y,X
   format], get data type from dictionary
      # note that band is a GDAL pointer.                                  96
                datarray = np.zeros(( band.YSize,band.XSize ), gdt2npy[
                   band.DataType])
                        # create loop based on YAxis (i.e. num rows)       98
                for i in range(band.YSize):
                        # read lines of band    using GDAL ReadRaster      100
                           function (xoffset, yoffset, cols, rows,
                           buf_xsize, buf_ysize, datatype)
                        scanline = band.ReadRaster( 0, i, band.XSize,
                           1, band.XSize, 1, band.DataType)
                        # unpack from binary representation using          102
                           Python struct module
                        # struct.unpack(datatype * nuumber cols, data)
                        # conversion between GDAL-Python datatype names   104
                            is done using RasterIO dictionary
                           gdt2struct
                        tuple_of_vals = struct.unpack(gdt2struct[band.
                           DataType] * band.XSize, scanline)
                        # tuple_of_floats = struct.unpack('f' * band.      106
                           XSize, scanline)
                        # add tuple to image array line by line
                        datarray[i,:] = tuple_of_vals                     108

# [return datarray]                                                       110
```

Listing 4.1: An extract from the RasterIO.readrasterband function showing reading a raster image file to a NumPy array.

The RasterIO module also supports raster scenes which contain pixels with null values. This is achieved using NumPy masked arrays. The masked array class is built on top of the standard NumPy array library and applies a no data type value to cells which match a no data value specified in the raster image meta-data. Masked rasters are only created if the input raster has an a 'NoDataVal' attribute as returned by the GDAL pointer. The standard NumPy array operators can then be used on the masked array, but with masked cells being ignored during numerical operations. This is useful when processing

raster scenes with null pixel values (e.g. from cloud cover) which could, for example bias a spatial average if they were included in the calculation. The creation of a masked array in RasterIO is the same as described above for a standard array but all values in the output array equal to the specified value are masked.

The `readrastermeta` function complements the `readrasterband` function and reads the geospatial raster meta-data from a raster file (Figure 4.3). Using `readrastermeta` with `readrasterband` means that when a raster is loaded into a NumPy array the geospatial information can be retained through the Python processing flow-line. The `readrastermeta` function takes a GDAL pointer from `opengdalraster` and returns the GDAL driver name (image format e.g. GeoTiff), number of columns, number of rows, details of image projection in Well Known Text format (WKT) and the coordinate reference information for the top left pixel of the raster (returned as a tuple of x,y coordinates, resolution and rotation).

Listing 4.2 shows an example of how the RasterIO read functions can be combined in a Python script to read thermal band 4 from an AVHRR scene, using the processing flow-line shown in Figure 4.3. In this instance once the data is read-in the script converts the band 4 values from degrees Kelvin to Celsius (Listing 4.2, Line 21). Note that the conversion takes advantage of NumPy's memory efficient vectorised array operations to perform the conversion using an optimised internal iterator which doesn't require any additional memory to store the output values. Once the conversion is complete the script prints some descriptive statistics about the scene using NumPy aggregate array functions (Listing 4.2, Line 25). Listing 4.3 shows the output from the script in Listing 4.2 including the geospatial meta-data attributes as well as summary statistics for AVHRR band 4.

```
import rasterIO as rio

# open a pointer to an AVHRR scene
# opengdalraster takes a filename as string.
gpointer = rio.opengdalraster('AVHRR_200308141056.tif')

# read thermal band 4.
# readrasterband(GDAL pointer, band number, no data value, masked flag)
b4 = rio.readrasterband(gpointer, 4, NoDataVal=9999, masked=False)

# Get the geospatial meta-data
driver, xsize, ysize, proj, geot = rio.readrastermeta(gpointer)
# Print the geospatial meta-data
print 'driver: ',driver
print 'xsize: ', xsize,
print 'ysize: ', ysize,
print 'proj: ', proj
print 'geot: ', geot

# Convert Kelvin values in band 4 to Celsius
b4 -= 273.15

# Now find some descriptive statistics of band 4 in this scene
print 'min, max, mean and std'
print b4.min(), b4.max(), b4.mean(), b4.std()

exit(0)
```

Listing 4.2: An example script using RasterIO to read AVHRR thermal band 4 into a NumPy array.

```
driver: GTiff

xsize: 1253

ysize: 1248

proj: PROJCS["OSGB 1936 / British National Grid",GEOGCS["OSGB 1936",
    DATUM["OSGB_1936",SPHEROID["Airy 1830"
    ,6377563.396,299.3249646000044,AUTHORITY["EPSG","7001"]],AUTHORITY[
    "EPSG","6277"]],PRIMEM["Greenwich",0],UNIT["degree"
    ,0.0174532925199433],AUTHORITY["EPSG","4277"]],PROJECTION["
    Transverse_Mercator"],PARAMETER["latitude_of_origin",49],PARAMETER[
    "central_meridian",-2],PARAMETER["scale_factor",0.9996012717],
    PARAMETER["false_easting",400000],PARAMETER["false_northing"
    ,-100000],UNIT["metre",1,AUTHORITY["EPSG","9001"]],AUTHORITY["EPSG"
    ,"27700"]]

geot: (-420172.78893345821, 1100.0, 0.0, 1146162.9352023469, 0.0,
    -1100.0)

min, max, mean & std
17.37, 29.0305, 24.4642354837, 1.86558708556
```

Listing 4.3: Output from Listing showing raster geospatial meta-data and band statistics.

To write NumPy arrays to new raster files the RasterIO module contains the functions `newgdalraster`, `newrasterband` and `writerasterbands` (Figure 4.4). To write a NumPy array to a raster file a GDAL pointer to a new file is opened by the `newgdalraster` function. This function takes the file name of the new file and the meta-data parameters for the output raster band. The meta-data parameters are the same as those returned from the aforementioned `readrastermeta` function (Figure 4.4) except that for convenience the projection information is represented as an EPSG code (European Petroleum Survey Group, EPSG (2012)). An EPSG representation of the projection information can be generated using the `wkt2epsg` utility function (see Appendix B). Next, the GDAL pointer for the new file returned by `newgdalraster` is

passed to the `newrasterband` function along with the NumPy array and the raster file data type. The `newrasterband` function then writes the NumPy array to a band in the new file. Once the write is complete the pointer to the file is closed and the new raster file is available to the user.

For convenience `writerasterbands` wraps both `newrasterband` and the `newgdalraster` methods into a single function (Figure 4.4). Additionally, the `writerasterbands` function supports writing multiple arrays as multiple bands to a new file (Listing 4.4) by iterating through the list of input arrays, calling the `newrasterband` function to write each array as a new band in the output file (Line 191).

```
# create function to write GDAL rasters from NumPy arrays                    182
def writerasterbands(filename, format, XSize, YSize, geotrans, epsg,
   NoDataVal=None, *rasterarrays ):
        ''' Accepts Numpy arrays, outputfile string, format and               184
           geotranslation metadata and writes to file on disk.'''
        # get number of bands
        num_bands = len(rasterarrays)                                         186
        # create new raster using newgdal raster(output filename, GDAL
           drive, cols, rows, geotranslation_params, EPSG_code,
           num_bands, output_datatype (using rasterIO npy2gdt dict).
        dst_ds = newgdalraster(filename, format, XSize, YSize, geotrans       188
           , epsg, num_bands, npy2gdt[rasterarrays[0].dtype.name])
        # add raster data from raster NumPy arrays
        band_num = 1 # band counter                                           190
        for band in rasterarrays:
                newrasterband(dst_ds, band, band_num, NoDataVal)              192
                band_num += 1
        # close output and flush cache to disk                                194
        dst_ds = None
```

Listing 4.4: The RasterIO writerasterbands function to write NumPy arrays to raster a image file.

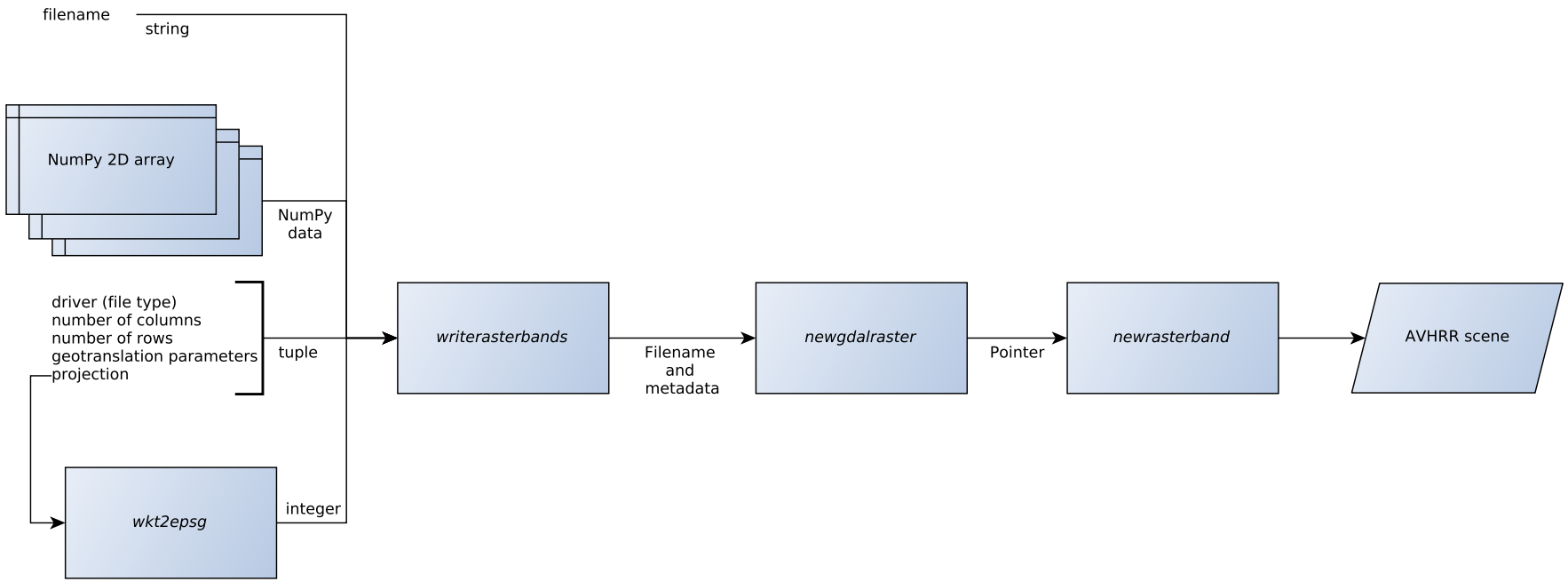


Figure 4.4: Data flow-line showing RasterIO functions to write NumPy arrays as raster images.

An example of using the RasterIO output functions is shown in Listing 4.5 where near-infrared bands 1 and 2 from an AVHRR scene are used to derive Normalised Difference Vegetation Index (NDVI) (Line 14). The NDVI array is then written to a new file using the geospatial meta-data from the input AVHRR images, using the process as shown in Figure 4.4 (Line 21).

```
import rasterIO as rio

# open a pointer to an input AVHRR scene.
gpointer = rio.opengdalraster('AVHRR_20030812.tif')

# read infra-red bands 1 and 2.
b1 = rio.readrasterband(gpointer, 1)
b2 = rio.readrasterband(gpointer, 2)

# Get the geospatial meta-data
driver, xsize, ysize, proj, geot = rio.readrastermeta(gpointer)

# Calculate NDVI ratio from bands one and two
ndvi = (b2-b1)/(b2+b1)

# Get the EPSG code for the projection
epsg = rio.wkt2epsg(proj)

# Write the output NDVI as a single band image
# Writerasterbands takes (filename, gdal_driver, cols, rows, geot, epsg
    and NumPy arrays).
rio.writerasterbands('ndvi_image.tif', driver, xsize, ysize, geot, epsg
   , ndvi)

exit(0)
```

Listing 4.5: An example script using RasterIO to read calculate NDVI and write the output to a new single band image.

4.2.2 PyAVHRR - a Python module for processing AVHRR data

Using the RasterIO module it is possible to develop high level functions to perform AVHRR specific tasks. The PyAVHRR module contains a series of functions used for pre-processing and analysis of AVHRR data using the RasterIO module. Appendix B, Section B.2 contains the documentation and complete source code for the PyAVHRR module.

The most important function in this module is the `lst_DLR` function, which implements the land surface temperature estimation algorithm developed by the German Aerospace Centre (DLR) (see Section 3.2.12). Listing 4.6 shows the `lst_DLR` function from the PyAVHRR module. The `lst_DLR` function takes AVHRR bands 4 and 5 (top of atmosphere brightness temperature) and NDVI derived from bands 1 and 2 in the same AVHRR scene, as NumPy arrays, and returns an array of estimated land surface temperature. Listing 4.6 shows that the NDVI inputs are scaled between 0 and 255 (representing a 256 bit image) (Line 82). The scaled NDVI values are then used to derive the emissivity coefficients (Lines 84-87). Estimated land surface temperature (EST) is calculated using a NumPy vectorised iterator to apply the DLR equation to each pixel in-turn, returning a NumPy array containing EST for the same pixels as captured in the original AVHRR scene (Line 88). Listing 4.7 shows an example of using the PyAVHRR `lst_DLR` function in a Python script.

In the LST script (Listing 4.7) the PyAVHRR `readbands` and `ndvi` utility functions are shown. The `readbands` function (Line 12) is a wrapper for the `RasterIO.readband` function, which reads each of the five AVHRR bands and returns a tuple of the arrays accessed through band numbers one to five. The `ndvi` function calculates NDVI from bands 1 and 2 and returns the result (Line 14). The NDVI value along with data from bands four and five is then passed to the `lst_DLR` function and the output LST array is then written to a new file (Line 19).

```
# Function to calculate estimated land surface temperature from AVHRR ch4/ch5 (see http://eoweb.dlr.de/short_guide/D-LST.html)
def lst_DLR (b4, b5, ndvi):
        '''Accepts a NDVI raster and AVHRR bands 4 and 5, derives a new raster of estimated land surface temperature (Kelvin).

        This function calculates estimated land surface temperature using a split window method to correct
        atmospheric attenuation of thermal bands and NDVI to apply a correct for surface emissivity.
        Returns a new single raster of Estimated surface temperature in Kelvin.

        Developed by Deutsches Zentrum fur Luft- und Raumfahrt (DLR) - German Aerospace Center
        Visit: http://eoweb.dlr.de/short_guide/D-LST.html for more details.

        >>> rlst = avhrr.lst_DLR(b4, b5, fndvi)
        '''
        # first scale any NDVI inputs
        ndvi255 = (ndvi+1)*127
        # second, calculate coefficients
        e4 = 1.0094 + 0.047*(np.log(ndvi255))
        e5 = e4 + 0.01
        e = (e4 + e5)/2
        de = e4 - e5
        lst_raster=1.274+(b4+b5)/2*(1+0.15616*((1-e)/e)-0.482*de/(np.power(e,2)))+(b4-b5)/2*(6.26+3.989*((1-e)/e)+38.33*de/(np.power(e,2)))
        return lst_raster
```

Listing 4.6: The PyAVHRR lst_DLR function to calculate estimated surface temperature from AVHRR bands 4 and 5 using the DLR method.

```
import RasterIO as rio
import PyAVHRR as avhrr

# Get GDAL pointer to scene
gpointer = rio.opengdalpointer('AVHRR_20030812.tif')
# Get geospatial meta-data
driver, xsize, ysize, proj, geot = rio.readrastermeta(gpointer)
# Get the EPSG code for the projection
epsg = rio.wkt2epsg(proj)

# Read all five AVHRR bands (wrapper for RasterIO functions)
scene = avhrr.readbands(gpointer)
# Calculate NDVI ratio from bands one and two
ndvi = avhrr.ndvi(scene[1],scene[2])

# Calculate LST
lst = avhrr.lst_DLR(scene[4],scene[5],ndvi)
# Write the output LST as a single band image
rio.writerasterbands('lst_image.tif', driver, xsize, ysize, geot, epsg,
    lst)

exit(0)
```

Listing 4.7: An example script using the PyAVHRR module to calculate EST from an AVHRR scene using the lst_DLR function.

4.2.3 Raster Processing Suite

The Raster Processing Suite is an extension to the RasterIO module which provides a graphical user interface for performing raster numerical processing within the Quantum GIS (QGIS) package (QGIS, 2012). The advantage of the processing suite is that raster scenes can be visualised in QGIS before and after processing, aiding exploratory analysis. The Raster Processing Suite provided the first "raster calculator" functionality in the QGIS package, prior to the inclusion of the internal raster calculator tool in version 1.6 (Sutton, 2010). The source code for the Raster Processing Suite is shown in

Appendix B, Section B.3.

The Raster Processing Suite contains two interfaces for manipulating raster data, the Processor and the Python script interface. The Processor (Figure 4.5) contains an equation editor which allows the user to develop equations for raster images using NumPy's vectorised processing functions. The user loads a raster band into memory using the "Load Band" button (Figure 4.5). This creates a NumPy array in the global memory space of the Raster Processing Suite, using the RasterIO read function to read the values from the file specified (Figure 4.3). The name of the array added to the Equation Editor is based on the name of the file and the band loaded. For example band 1 from 'avhrr_file.tif' would be 'avhrr_file_1'. Once a band is loaded it is possible to use the Python `globals()` dictionary, with the array name acting as the dictionary key, to retrieve the array data. If the user loads or refers to the same band more than once, only one instance of the band is created in memory.

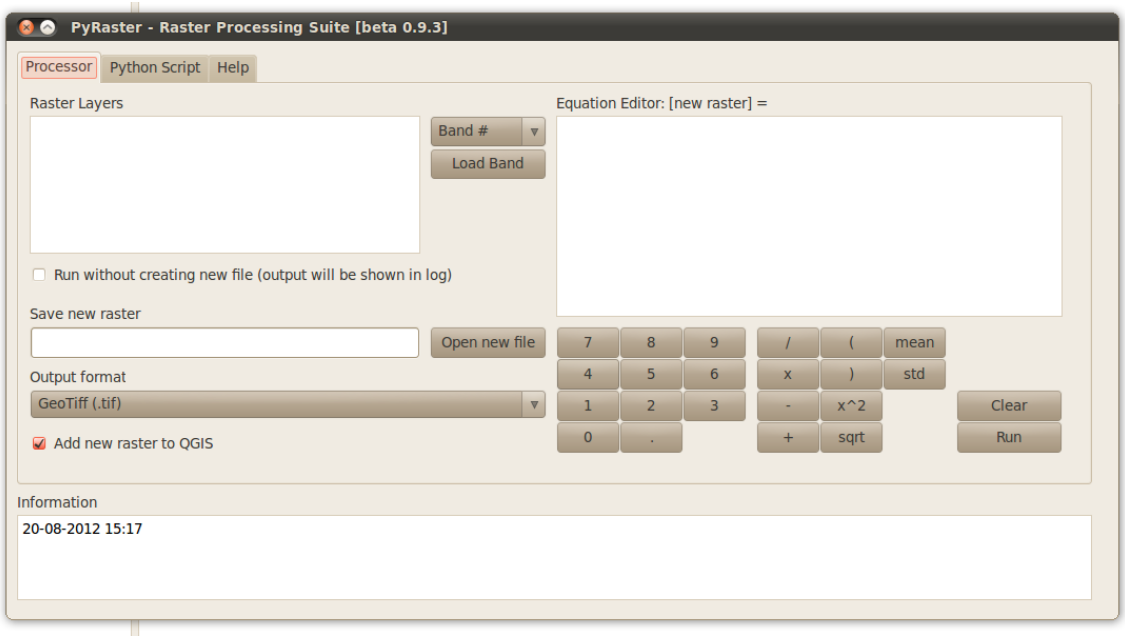


Figure 4.5: Screen-shot of the Raster Processing Suite in Quantum GIS.

The equation editor accepts standard NumPy arithmetic for processing operations. The user selects an output file and image format for the result. Alternatively, the output can be printed to the console, a useful feature when performing exploratory analysis using spatial aggregates (e.g. an average of all pixels). Once the output is defined the user can press the 'Run' button

and the Raster Processing Suite will attempt to process the equation in the Equation Editor using the loaded raster bands. The equation is converted to a plain string, and parsed using the Python `eval` function which interprets the string as a Python expression. References to the loaded raster arrays are detected automatically as they exist in global memory space. The output from the equation is returned as a NumPy array, which is written to a new file on disk if specified by the user. Figure 4.6 shows the Equation Editor being used to generate NDVI from bands 1 and 2 in an AVHRR scene, replicating the process shown in Listing 4.5.

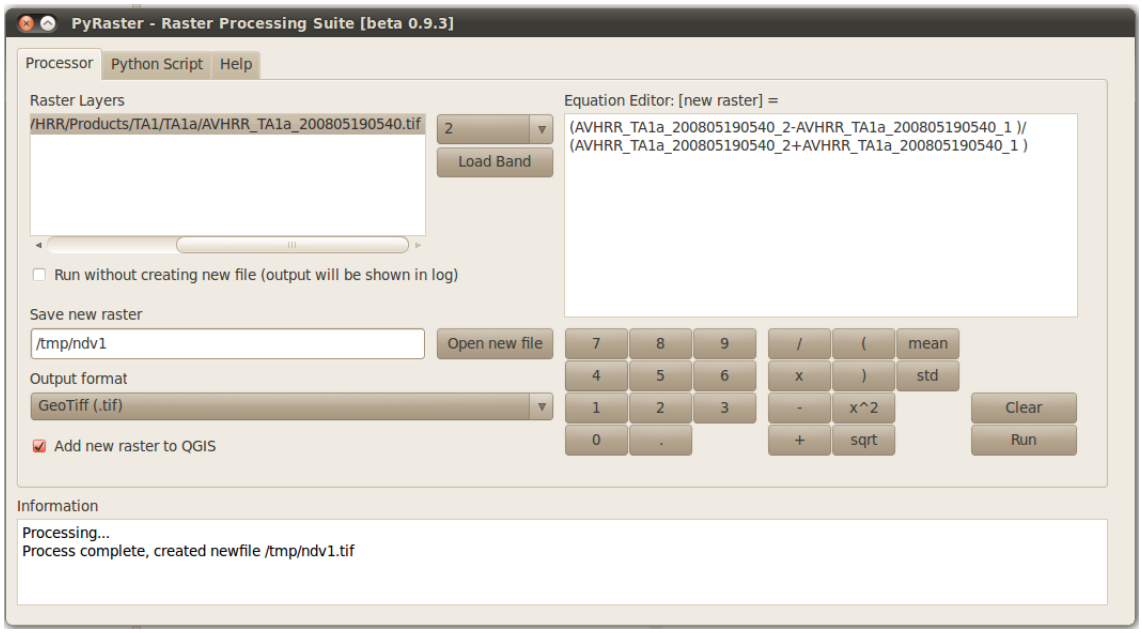


Figure 4.6: Screen-shot of the Raster Processing Suite in Quantum GIS being used to calculate the NDVI using an AVHRR scene.

The Scripting Interface provides a text editor window to create Python scripts for raster processing using the RasterIO and PyAVHRR modules. The code entered into the Scripting Interface is executed as a stand-alone Python process and can be be run within the Raster Processing Suite. The interface is dynamically linked to the Equation Editor window so that the user may develop a process using the graphical user interface in the Equation Editor and view the process as an automatically generated Python script. The script can then be saved as a standard Python file and used at a later date.

The purpose of the Scripting Interface is to enable rapid development of Ras-

terIO processing routines. For example, the user could develop a raster process using a single scene in the Equation Editor and then use the Python representation in the Scripting Interface to apply the process iteratively over a time-series of scenes. Figure 4.6 shows the Scripting Interface with an automatically generated script from the NDVI calculation in the Equation Editor window.

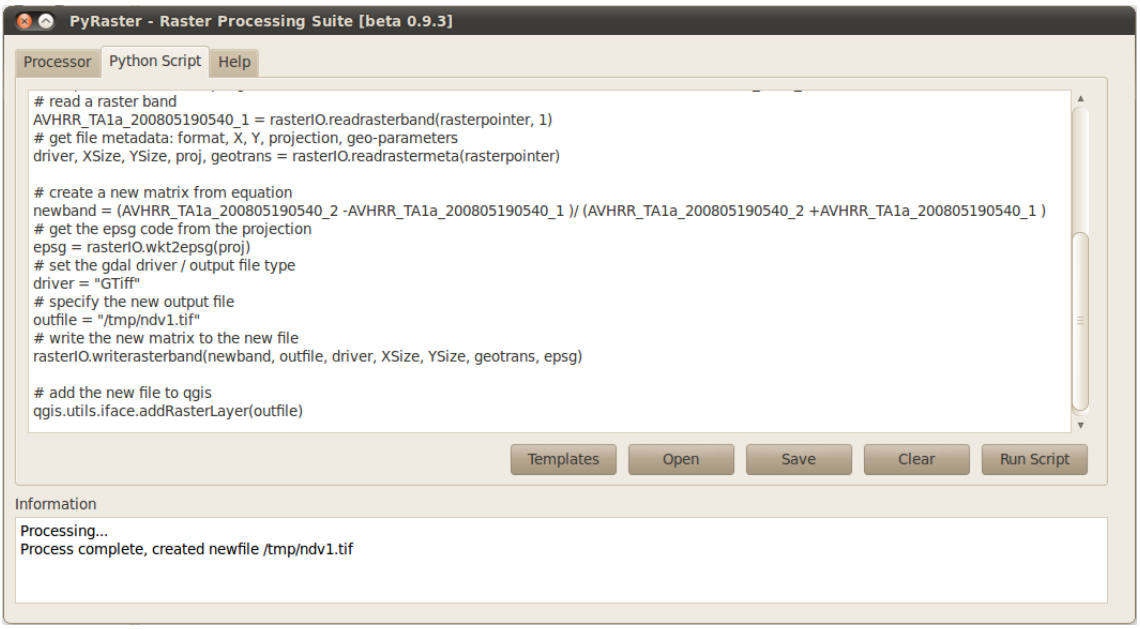


Figure 4.7: Screen-shot of the Raster Processing Suite in Quantum GIS showing the auto-generated script to calculate NDVI using an AVHRR scene.

4.3 Database interface

As discussed previously in Section 3.2.7 to effectively manage the catalogue of time-series AVHRR scenes a meta-data PostGIS database was created containing AVHRR scene meta-data attributes. Alongside this database the MIDAS PostGIS database was created containing the time-series of weather station measurements. Using these databases measurements from AVHRR and MIDAS data were indexed by time and location, enabling spatio-temporal analysis of temperatures across London.

The database interface layer provides the framework with access to the PostGIS tables containing the AVHRR meta-data and MIDAS measurements. The

database interface uses three libraries, each performing different functions: the Psycopg module is an external library (see Table 4.1) which is a PostgreSQL adaptor for Python. Psycopg was used in the framework to develop a conversion between PostgreSQL tables and NumPy structured arrays. The RPy module is the Python interface to the R statistical language and can be used to access the RPostgreSQL adaptor (see Table 4.1), which converts data between PostgreSQL and R data types. This was used during development of statistical routines as it allows conversion of PostgreSQL/PostGIS data into the R data frame data-type, nested within Python, suitable for testing statistical analysis. The CSQL library was constructed as part of the framework during the data pre-processing (Section 3.1) to execute queries on the command line and return large volumes of data from PostGIS to Python and Bash scripts for batch data processing.

The creation of a spatio-temporally paired data-set of AVHRR EST and MIDAS air temperatures (see Section 5.2.2) was achieved by extracting values for pixels in AVHRR EST scenes and writing these values to a new table in the PostGIS MIDAS database. This process relied on the database interface, and is presented here as an example of database connectivity within the developed framework. The complete source code for the script to extract AVHRR pixel values is presented in Appendix H. Figure 4.8 shows the flow-line of processes for the extraction of pixels in the script.

The process for extracting pixel values can be divided into five sub-processes (Figure 4.8), with a database connection defined using the Psycopg library. The selection of scenes for use (see Section 5.2.2 for selection criteria) was achieved using an SQL query (see extract of source code, Listing 4.8, Line 94). The query to select AVHRR scenes performs a temporary join between the metadata table ('order_avhrr_1985_2008_cloud0' and the file names table ('avhrr_ta2a') based on scene primary key defined during the AVHRR pre-processing (see Section 3.2 and Figure 3.4). The query then selects the file names of daytime scenes (06:00-21:00) and returns a list of scenes (primary key, filename, satellite, date and time) ordered by date and time. The query was executed by the database using the Psycopg module (Line 95), and re-

turned a list of file names for the selected AVHRR EST scenes (Line 96).

```
# Connection to database
conn = psycopg2.connect(database=opts.indatabase, user=opts.user, host=
    opts.server, password=pgs)
# Open database cursor
cur = conn.cursor()
# Get list of files to process from database
SQL = "SELECT t.primary_key, t.fname, o.satellite, o.date_of_scene, o.
    time_of_scene FROM avhrr_ta2a t, order_avhrr_1985_2008_cloud0 o
    WHERE o.primary_key = t.primary_key AND o.time_of_scene >=
    '06:00:00' AND o.time_of_scene <= '21:00:00' ORDER BY o.
    date_of_scene, o.time_of_scene;"
cur.execute(SQL)
fdata = cur.fetchall()
```

Listing 4.8: Extract of source code (see Appendix H) for extraction of AVHRR EST pixel values showing selection of scenes using the AVHRR meta-data database.

To speed-up the pixel value extraction process over the time-series the Python Multiprocessing module functions `Queue` and `Process` were used to execute multiple process threads for the extraction of pixel values simultaneously (Figure 4.8 (2)). The number of threads was determined by the number of processors available on the host machine. Given two processors, the file name list is divided into two, and each processor was used to extract pixel values from half of the file list (Listing 4.9).

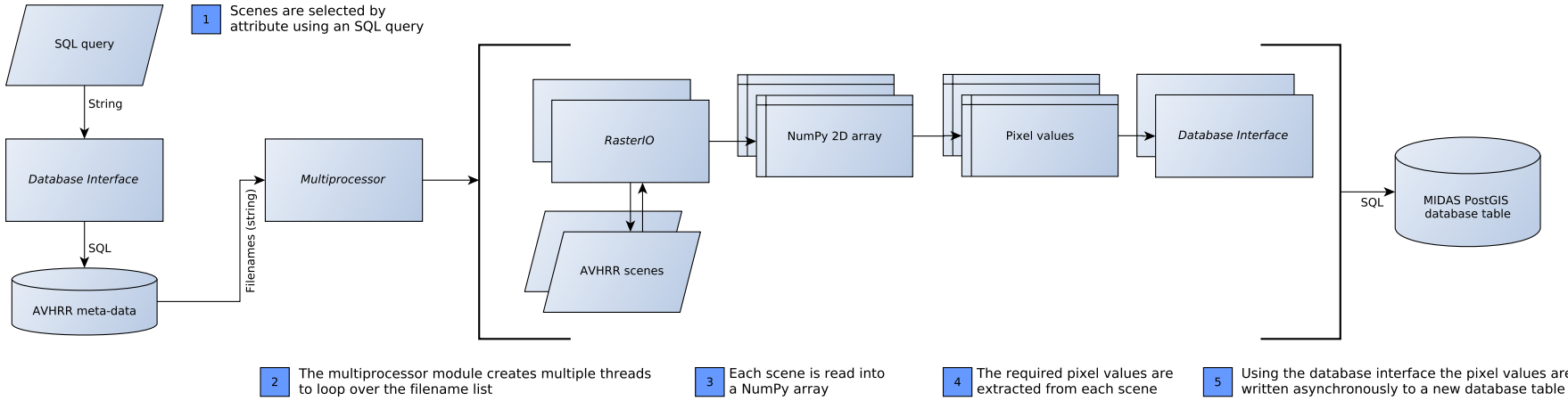


Figure 4.8: Data flow-line showing the extraction of AVHRR EST values from raster files to a PostGIS database table.

```
# Create a queue to run processes                                          126
q = Queue()
p1 = Process(target=processRun,args=(q,fdata[:length],opts.outdatabase,  128
    opts.user,opts.server,pgs))
p2 = Process(target=processRun,args=(q,fdata[length:],opts.outdatabase,
    opts.user,opts.server,pgs))
# Start processes (dump data to table)                                     130
p1.start()
p2.start()                                                                 132
```

Listing 4.9: Extract of source code (see Appendix H) for extraction of AVHRR EST pixel values showing multiple computation threads using the Multiprocessing module.

To extract the pixel values, each of the files in the file name list was read into a NumPy array using the RasterIO module. The values for the required pixels were extracted based on row and column number (Figure 4.8 (3,4)). Listing 4.10 shows the `getpixelval` function which was created for this purpose.

```
def getpixelval(raster, Xpix, Ypix):                                      28
        '''Accepts Np raster and returns value of specified pixel'''
        return raster[Ypix,Xpix]                                           30
```

Listing 4.10: Extract of source code (see Appendix H) showing the getpixelval function for the extraction of AVHRR EST pixel values.

Once the required pixel values were extracted these were written to a new table in the PostGIS MIDAS database using SQL insert statements with the Psycopg library (Figure 4.8 (5)). Listing 4.11 shows an extract from the source code (Appendix H) of the SQL database insert for EST values from the pixel containing the London Weather Centre station. After the process was completed, the new MIDAS table held the AVHRR derived EST values for the four London weather stations selected for the analysis of EST and air temperatures (see Section 5.2.1).

```
# Listing showing extract of avhrr2pgdb - inserting EST for LWC via a
   Psycopg connection.
# Note, cur is a PostgreSQL database connection cursor from the Psycopg
    library.
# 19144 - LHR
                        cur.execute("INSERT INTO est_gla_avhrr_t (
                           primary_key, fname, satellite,
                           date_of_scene, time_of_scene, src_id, est)
                           VALUES (%s, %s, %s, %s, %s, %s, %s);",((
                           pkey, fname, sat, date, time, 19144, lwc)))
                        #print pkey, fname, sjp, lhr, nth, kew, lwc
```

Listing 4.11: Extract of source code (Appendix H) showing the insertion of EST values for London Weather Centre into the database.

4.4 Summary

The framework developed in this project provides a uniform data model using NumPy arrays for the analysis of AVHRR and MIDAS time-series data-sets. Using the NumPy array type both raster and vector data-types can be represented in compatible array structures, allowing integration of both data for analysis. Furthermore, the use of NumPy arrays provides optimised methods for array iteration, making processing large volumes of data in this format computationally efficient.

The management of both data-sets outside the data model is achieved using a spatial database, allowing the dynamic selection of data for processing by location and time, as well as management of meta-data through the processing flow-line. However, the success of the data model and database strategies is dependent on the framework's ability to perform translations between files/-database tables and NumPy arrays.

The Python language provides a suite of external libraries to help perform these data conversions, which are leveraged by the framework to enable communication with the spatial database, as exemplified in the pixel extraction

routine shown above. The RasterIO module provides an operational interface using the GDAL library to convert between raster files and NumPy arrays to perform processes over the time-series of AVHRR data. The RasterIO and PyAVHRR modules are used extensively within the framework for analysis (see Chapters 5 and 6) and also in external projects seeking to use long-time series AVHRR data (e.g. Hardy et al., 2012).

Chapter 5

Quantifying and modelling the relationship between urban surface and air temperatures

5.1 Introduction

Meteorological and climatological studies have traditionally used terrestrial observations of near-surface air temperature for analysis of the thermal environment of urban areas (Eliasson and Svensson, 2003; Oke, 1987; To et al., 2011). Air temperature is a fundamental variable for quantifying and modelling urban temperatures and human exposure to heat in cities (Balling and Brazel, 1987; Eliasson and Svensson, 2003; Kolokotroni and Giridharan, 2008; Ren et al., 2007). Furthermore, it has been recognised that the measurement of urban temperatures at the sub-city spatial scale (e.g. across different neighbourhoods) is required to develop our understanding of urban temperature dynamics in order to quantify the spatial variability of exposure to heat during extreme temperature events (Harlan et al., 2006; Rosenzweig et al., 2011; To et al., 2011).

However, as previously discussed in Section 2.4.4, near-surface urban air temperature measurements are often spatially sparse and hence are unable

to capture the intra-urban spatial temperature characteristics across an entire city (Dousset et al., 2011; Prihodko and Goward, 1997; To et al., 2011; Vogt et al., 1997). Earth observation offers spatially-complete and repeatable surface temperature data which has been widely used in urban climatology as an alternative to spatially-lacking terrestrial observations (Dousset and Gourmelon, 2003; Dousset et al., 2011; Forster, 1985; Fung et al., 2009; Gallo et al., 1995; Matson et al., 1978; Shepherd et al., 2002; To et al., 2011). Given the requirements for near-surface air temperatures (see Section 2.4.4) and the advantages of Earth observation, as discussed in Section 2.4.4, a number of studies have developed models for estimating spatially complete air temperatures from Earth observed surface temperatures based on empirically derived relationships between the two (Fung et al., 2009; Kawashima et al., 2000; Prihodko and Goward, 1997; To et al., 2011; Vogt et al., 1997).

However, previous urban-based studies have been criticised for their limited temporal coverage, as a result of using a small number of scenes (Voogt and Oke, 2003; Nichol et al., 2009; To et al., 2011). This is because thermal images only provide data for one point in time (To et al., 2011), yet temperature is a temporally continuous variable, and the relationship between estimated surface temperatures (EST) and air temperature varies at the sub-diurnal level (Vogt et al., 1997). To et al. (2011) showed that correlations between daytime EST (derived from two ASTER scenes) and in-situ air temperature were only significant at the 5% level up to one hour before and two hours after satellite overpass time. In contrast, the night time correlation was valid at the 5% level up to 5 hours before and after the satellite overpass. To et al. (2011) derived predicted air temperature based on the relationship between EST and in-situ air temperatures, and examined the correlation between predicted and measured air temperatures on days before and after the satellite overpass. The study showed that the r^2 value between predicted and measured air temperature could reduce by up to 0.55 within a 24 hour period as a result of changing weather conditions. As such, the work by To et al. (2011) and others (e.g. Prihodko and Goward, 1997; Vogt et al., 1997; Fung et al., 2009) shows that to derive temporally-robust empirical relationships

between EST and air temperatures, a long-time series of data with multiple scenes per day should be used to properly capture spatio-temporal variability in the EST-air temperature relationship.

The work presented in this chapter addresses this issue by using the long-time series of integrated daytime MIDAS air temperature measurements and AVHRR EST data (Sections 3.1 and 3.2) to analyse the relationship between daytime EST and air temperatures in the urban environment. Two key objectives are investigated: The first is to attempt to quantify the relationship between summer EST and air temperatures in London over the available time-series from 1985 to 2008, and examine whether location is a controlling factor on the strength of relationship. This will allow, as the second objective, an evaluation of whether the derived relationship can be used to model air temperature as a function of surface temperatures over the entire Greater London area. These objectives will enable an evaluation of whether satellite-based empirically modelled air temperature can be used as a proxy for terrestrial air temperature measurements in order to capture spatially complete daytime urban temperature dynamics over a multi-decadal time series.

Whilst the utility of thermal Earth observed ESTs to capture intra-urban spatial variability of the heat hazard during heatwaves has been demonstrated (Dousset et al., 2011), air temperature is required to quantify human heat-stress (Baker et al., 2002; Harlan et al., 2006) in order to model changes in the risk of increased morbidity and mortality of vulnerable populations during heatwaves (Baker et al., 2002; Johnson et al., 2005; Harlan et al., 2006; Kovats et al., 2006). Therefore, spatially-complete modelled air temperatures may be used to better understand spatial variability to increased exposure of vulnerable populations to the heat hazard during extreme temperature events, providing invaluable information for future mitigation and adaptation options (Solecki et al., 2005; Rosenzweig et al., 2011).

5.2 Data pre-processing

To enable analysis of air and surface temperatures it was necessary to combine the data from the MIDAS database and AVHRR derived EST raster images. EST and air measurements were paired spatially and temporally so that a like-for-like comparison of the two data-sets could be undertaken. The pre-processing was carried out in three steps; first, suitable weather stations were identified in the MIDAS data, second EST corresponding to the location of the selected stations were extracted from the raster (AVHRR) data and last, EST and air temperatures were paired by station and observation time and entered as a new table in the MIDAS database (Figure 5.1(a)). The following subsections describe these steps in detail.

5.2.1 Selection of MIDAS weather stations

For analysis of air and surface temperatures four weather stations measuring hourly screen-level air temperature within the Greater London Authority (GLA) were selected from the post-processed MIDAS database. The stations selected were: St James's Park (SJP), London Heathrow (LHR), Northolt (NTH) and London Weather Centre (LWC) (Table 5.1). These stations were selected based on the number of hourly daytime (08:00-21:00) temperature observations available for summer months (June-August) between 1985 and 2008. Table 5.1 shows the selected stations and their key attributes including the number of observations and time-series percent completeness. The table shows that missing data and duplicate observation removal during the MIDAS data pre-processing (Section 3.1) resulted in a high-percentage of missing observations at SJP and NTH, both of which have only approximately 50% completeness. In the case of SJP removal of duplicates from the Metform 3208 observation group (monthly returns of daily observations) during MIDAS pre-processing resulted in no data being available prior to 1995 (Figure 5.2).

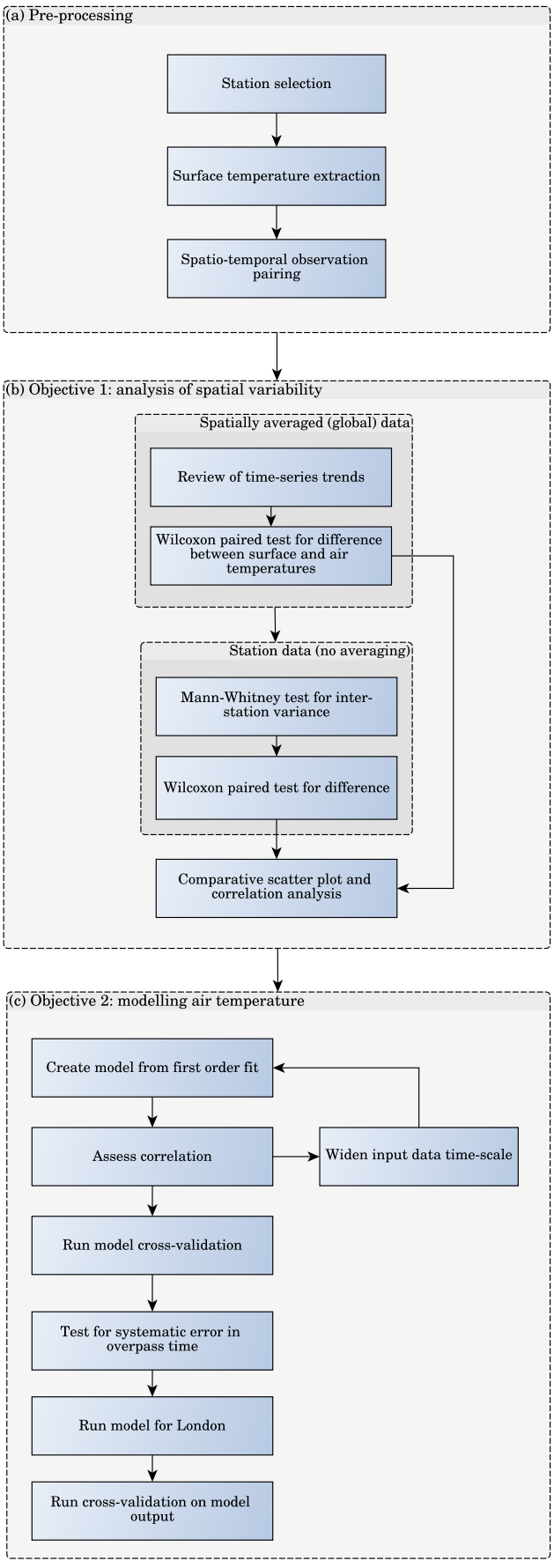


Figure 5.1: Flow-line of pre-processing (see Sections 5.2.1, 5.2.2 and 5.2.3) and analysis (see Sections 5.3.1 and 5.3.2) methodologies, including statistical tests used for analysis.

In this regard it is clear that the pre-processing strategy had a detrimental effect on data availability at SJP and NTH. However, the volume of observations in the original MIDAS data meant that to ensure the highest quality data-set possible, a data-wide filtering method had to be employed, as examination and filtering based on individual stations would have been too time consuming (see Section 3.1). Despite their low percentage of time-series completeness SJP and NTH were retained for analysis in the interests of a greater spatial coverage and diversity of land cover types as these stations represent urban green-space (SJP) and urban/vegetation mix (NTH). The four stations selected represent the best quality hourly time-series observations within the Greater London Authority (GLA) area.

Figure 5.2 plots the availability of observations over summer months since 1st June 1985 and shows that there are a greater number of observations later in the time-series, as the availability of observations in the MIDAS data archive improved. This weighting in the availability of data in the time-series could lead to a bias when looking at trends in temperature over time. However, the key objective of the study was analysis of the relationship between EST and air temperature using spatially and temporally paired observations. Observations were only compared when they existed in both air and surface data. Moreover, pairwise tests were used where applicable to minimise bias from changes in observation density throughout the time-series.

Station	**Location (E N)**	**Elevation (m)**	**Start Date**	**End Date**	**Summer Observations (of potential 30,912.)**	**Summer Time-Series Complete (%)**
LWC	531051 181981	43	1985-01-01	2008-12-31	27,188	87.95
SJP	529808 180081	5	1995-09-05	2008-12-31	15,442	49.95
LHR	507693 176721	25	1985-01-01	2008-12-31	27,426	88.72
NTH	509866 184494	40	1985-07-03	2006-08-31	16,450	53.22

Table 5.1: Selected London weather stations and hourly observation attributes for daytime measurements (08:00 to 21:00) during summer months (1st June to 31st August). *Stations; LWC: London Weather Centre, SJP: St James's Park, LHR: London Heathrow, NTH: Northolt.*

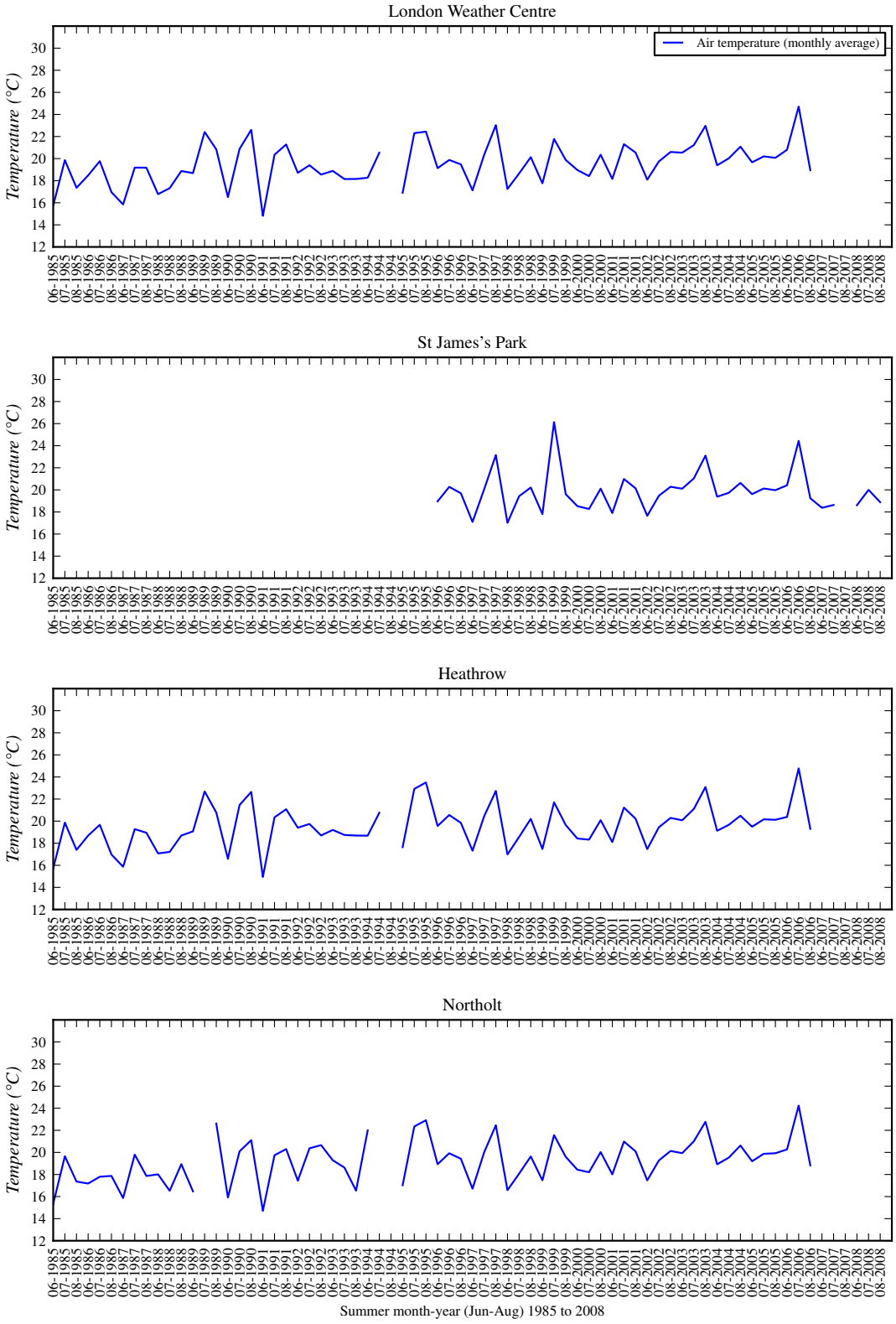


Figure 5.2: Air temperature measurement availability at the selected London weather stations for summer months (June-August) over the time-series (1985-2008).

5.2.2 Extraction of surface temperature observations

In order to conduct an analysis between EST and air temperatures at the selected weather stations within the study area it was necessary to integrate MIDAS air temperature observations with EST derived from the AVHRR scenes. This was achieved by extracting EST values from the pixels of scenes which contained the selected weather stations, and using these values to create a new table within the MIDAS database. This process is shown in Figure 5.3. A technical discussion of the software developed for this process is included in Section (4.3) and the complete source code is available in Appendix H.

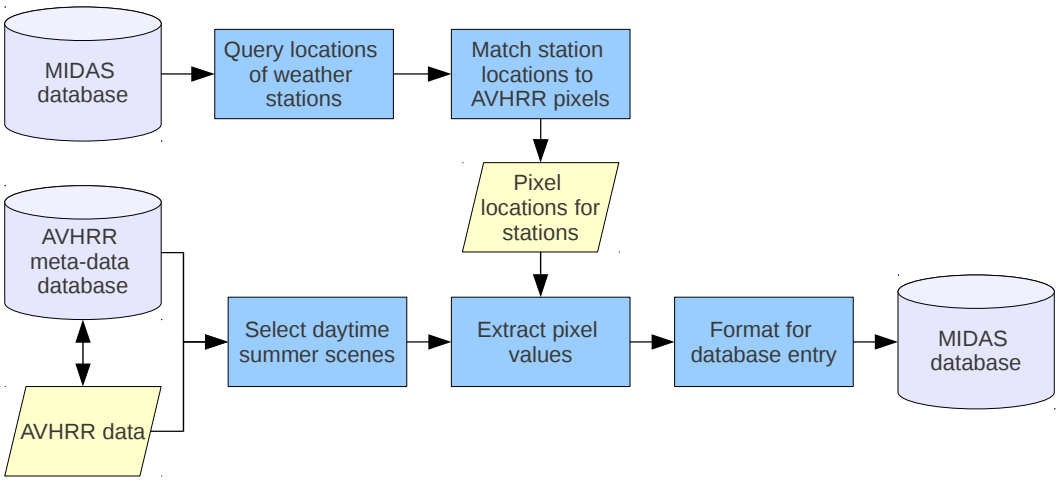


Figure 5.3: Pre-processing flow-line showing the extraction of EST pixel values for London weather station locations and their entry into the MIDAS database.

As described in Section 3.2 AVHRR daytime scenes were processed to EST with meta-data attributes stored in the AVHRR meta-database. Using the meta-database a list of file names of all available EST scenes that were captured from 1st June to 31st August between 08:00 and 21:00 over the time series was created. Within the archive there were 349 scenes available for the specified summer periods giving a total of 1396 possible surface temperature observations for all four selected stations.

The pixel coordinates (i.e. row and column numbers) of the selected terrestrial

weather stations were gathered using Quantum GIS by overlaying station locations from the MIDAS spatial database on one of the AVHRR scenes and manually selecting each of the four pixels that contained the selected weather stations. As described in Section 3.2, AVHRR pre-processing by Dundee Satellite Receiving Station geo-rectified the scenes to the same coordinate reference system and cropped them to the same extents, thus the pixel coordinate values for the four stations were valid for all scenes within the archive. Figure 5.4 highlights the location and pixel of the four selected stations in the GLA .

Using the RasterIO module (see Section 4.2) a Python script was developed to read each of the scenes in turn, extract the pixel values representing the EST for each of the four pixels and write these to a new surface temperature table in the MIDAS database (see Appendix H for script source code). Weather station ID and scene meta-data were recorded against observations from each of the four pixels so that surface temperature values could be referenced by station ID and origional AVHRR scene. Table 5.2 shows an extract of surface temperatures and attributes in the database table.

Scene Primary Key	**File Name**	**Satellite**	**Scene Date**	**Scene Time**	**Station ID**	**EST (°C)**	**Table Primary Key**
1795430912	AVHRR_TA2a_ 200806050954.tif	NOAA-17	2008-06-05	09:54	LWC	22.2	3404
18113915724	AVHRR_TA2a_ 200806081139.tif	NOAA-18	2008-06-08	11:39	SJP	20.73	3405
18113915724	AVHRR_TA2a_ 200806081139.tif	NOAA-18	2008-06-08	11:39	LHR	22.38	3406
18113915724	AVHRR_TA2a_ 200806081139.tif	NOAA-18	2008-06-08	11:39	LWC	21.76	3407

Table 5.2: An extract from the MIDAS database table containing AVHRR ESTs for the four London weather station locations. *Note: station numbers replaced with station letter codes for clarity. Stations; LWC: London Weather Centre, SJP: St James's Park, LHR: London Heathrow, NTH: Northolt.*

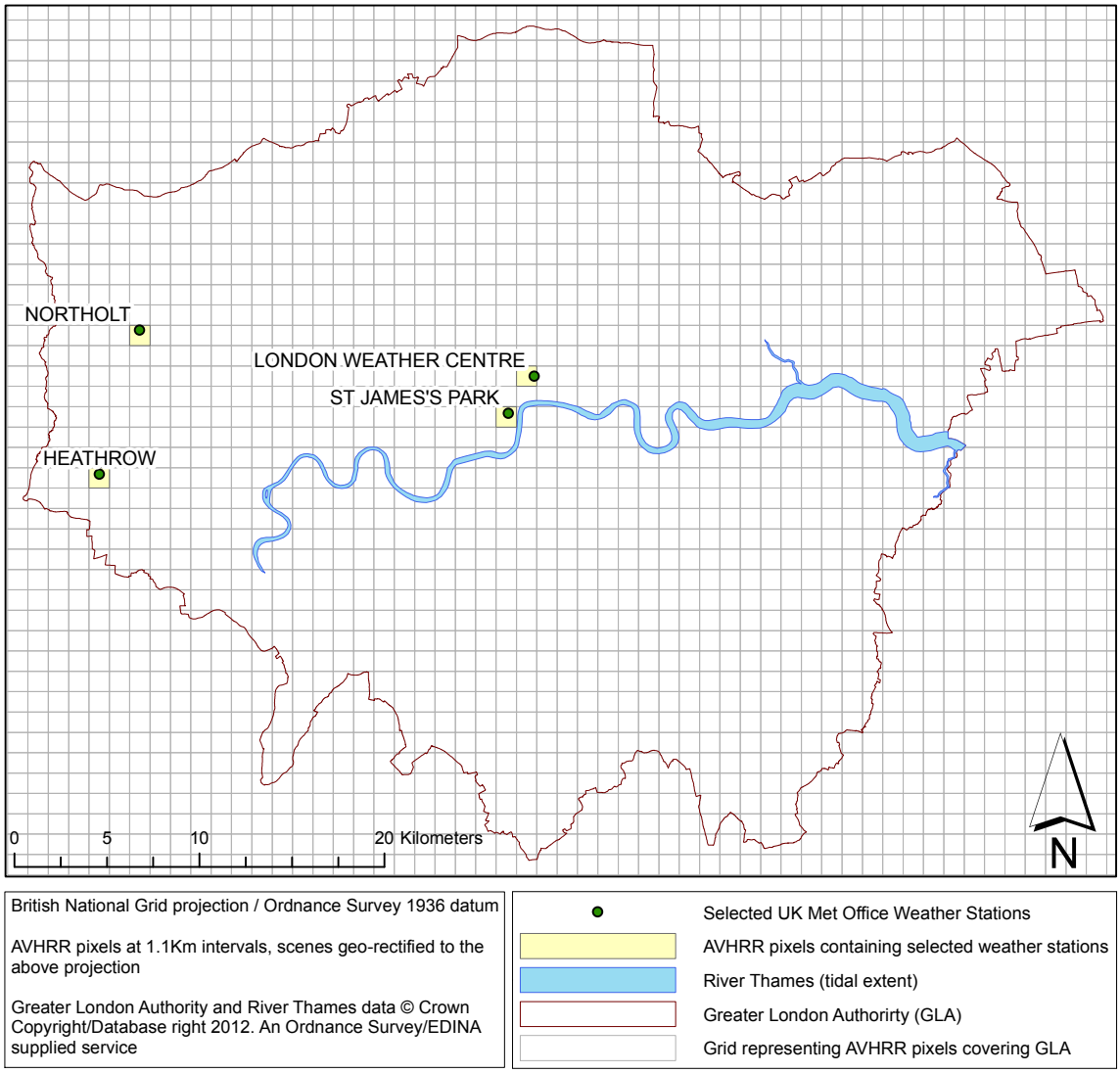


Figure 5.4: Location of selected weather stations and boundaries of AVHRR pixels within the Greater London Authority area.

5.2.3 Spatio-temporal pairing of air and surface temperatures observations

To create a temporally consistent data-set for each of the stations EST and air temperature measurements were paired using the weather station hourly observation time closest to the time of satellite overpass. For example if the satellite overpass of London was at 11:39 then the air temperature recorded at 12:00 would be paired with the EST at 11:39. Paired EST and air temperature measurements were then entered into a new MIDAS database table. If either EST or air temperature measurements were missing for the specified

time as a result of gaps in the MIDAS time-series (see Section 5.2.1) or cloud contaminated AVHRR scenes (see Section 3.2.8) then no record was written to the new table. Records were paired spatially by joining on station ID, so that paired measurements represented EST and air temperature from the same station location within a maximum of ±30 minutes of each other.

This process was automated using an SQL query to select and pair the temperature observations in a new table, and was repeated for each of the selected stations so that observations were paired temporally and spatially (Figure 5.5). Table 5.3 shows an extract from the database table. In the table the original AVHRR scene primary key was retained through the pairing so that EST measurements could be linked back to their satellite scene if required. Once this process was complete, the new table consisted of 1,110 paired EST and air temperature observations for the selected stations over the time-series duration. Of the original 1,396 EST measurements available from the four weather station locations, 286 could not be used because of discontinuity in the MIDAS data meaning that no air temperature record was available for pairing.

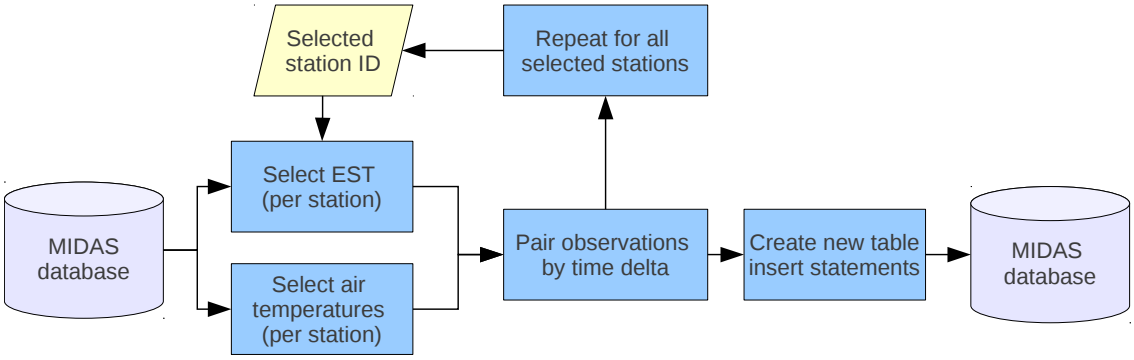


Figure 5.5: Pre-processing flow-line showing the spatio-temporal pairing of EST and air temperature measurements.

AVHRR Scene Primary Key	Scene Date	EST Time	Air Time	Δ (min)	Station ID	EST (°C)	Air Temp (°C)	Table Primary Key
18121517304	2008-09-28	12:15	12:00	15:00	NTH	13.76	17.00	2116
17102832549	2008-09-28	10:28	10:00	00:28	LHR	12.52	15.00	1627
17102832549	2008-09-28	10:28	10:00	00:28	LWC	11.14	14.70	586
17102832549	2008-09-28	10:28	10:00	00:28	SJP	10.70	15.30	945

Table 5.3: An extract from the MIDAS database table containing spatio-temporally paired EST and air temperature measurements (air temperatures are within 30 minutes of satellite overpass time (EST time)). *Note station numbers replaced with station letter codes for clarity. Stations; LWC: London Weather Centre, SJP: St James's Park, LHR: London Heathrow, NTH: Northolt.*

5.3 Analysis Methodology

The analysis of EST and air temperatures comprised five sections, divided into two groups, each corresponding to one of the objectives previously described in Section 5.1. Figure 5.1 shows the analysis methodology work flow for each objective. Section 5.3.1 describes the experiments for the first objective, to ascertain whether there was a quantifiable relationship between EST and air temperatures, and whether location was a driver of the strength of the relationship (Figure 5.1 (b)). Section 5.3.2 discusses the methodology employed to derive an empirical model of air temperature based on the EST-air temperature relationship (Figure 5.1 (c)). Section 5.3.3 describes the cross-validation testing used to evaluate the model, while Section 5.3.4 describes tests used to examine possible systematic errors in the EST-air temperature relationship and their potential influence on the model. Finally, Section 5.3.5 presents the methods used to generate spatially-complete estimates of air temperature using the model, and evaluate the outputs.

5.3.1 Quantifying the air-surface temperature relationship

To establish whether there was a "global" relationship between EST and air temperature independent of station location a mean average of EST and air temperatures from all four of the stations was calculated for each satellite overpass (termed 'spatial average').The spatial average was derived from the spatio-temporally paired temperatures database table (Table 5.3) by taking the mean of the EST and air temperature over the four weather station locations for each satellite overpass. Thus, for each date/time recorded in the database an averaged record of EST and air temperature was created.

Using the spatio-temporally paired data to generate spatial averages from the four stations meant that a valid observation of both EST and air temperature must have existed at a specified station location for that station to be included in both the EST and air temperature spatial averages. This ensured that both EST and air temperature spatial averages were representative of the same stations at the same point in time.

To examine differences in the long-term trends of the EST and air temperature data, the spatial average EST and air temperatures were temporally averaged to the monthly level, and plotted with first-order trend lines. The differences in the trend lines were used to qualitatively evaluate whether EST and air temperatures exhibited similar trends over the time-series. To evaluate whether the spatial averages of EST and air temperature data were statistically different a two-tailed pair-wise non-parametric Wilcoxon signed rank test at the 95% confidence level from the R statistical package was used. The null hypothesis (H_0) for this test was that there was no significant difference between spatial averages of EST and air temperatures. The alternative hypothesis (H_1) asserted that there would be a significant difference between the spatial averages of EST and air temperatures. Whilst a correlation based test could have been used to assess the EST-air temperature relationship the Wilcoxon difference test was chosen for use in this experiment to test the hypothesis without a priori assumptions on the relationship between EST and

air temperatures, in a distribution-free and pair-wise manner.

After testing the spatial averages of EST and air temperature data, the full suite of EST and air temperatures from all four stations from each of the stations were tested. First, to examine whether either EST or air temperatures exhibited significant spatial variability between station locations, a two-tailed Mann-Whitney U-test at the 95% confidence level was used to test for distribution differences in EST and air temperatures between the stations. A pair-wise test could not be employed due to the number of measurements varying at different stations (e.g. missing data at SJP) so the Mann-Whitney test (also non-parametric) was used. The null hypothesis (H_0) for testing of EST stated that there was no significant difference between EST recorded at the different stations. The alternative hypothesis (H_1) stated that there was a significant difference in EST between the different stations. Similarly, for the air temperatures the null hypothesis (H_0) stated that there was no significant difference in air temperatures between the different stations. The alternative hypothesis (H_1) was that there was a significant difference in air temperatures between the different stations.

Next, to assess whether there was spatial variation in the relationship between EST and air temperature at each of the stations at the 95% confidence level, a two-tailed Wilcoxon test was used. The null hypothesis (H_0) asserted that there was no significant difference between EST and air temperatures at the different stations. The alternative hypothesis (H_1) was that there was a significant difference between EST and air temperatures at the different stations.

In the final section of EST-air temperature analysis prior to development of an empirical model, the effect of location on the EST-air temperature relationship was investigated by comparing correlation coefficients of EST and air temperatures from the full suite of data from all stations, and correlation coefficients between the spatial averages of EST and air temperature. Scatter plots of the two EST-air temperature data-sets (full suite and spatial averages) were created, with a line of best fit to help identify changes in

the strength of the relationship as a result of the spatial averaging procedure described above.

5.3.2 Modelling the air-surface temperature relationship

The initial analysis of the EST and air temperature relationship and spatial variability (Section 5.3.1) was undertaken with MIDAS and AVHRR data constrained to the summer months June-August for daytime hours 08:00-21:00 (see Section 5.2). However, the MIDAS and AVHRR data includes a further 1010 measurements from the months of May and September for daytime hours 06:00 to 21:00 (Section 3.2).

Therefore, it was decided to include this data in the modelling stages so that the empirical models generated would be representative of a greater summer temporal and diurnal range. Furthermore, this provides a greater number of samples for cross-validation of the models. The pre-processing steps described in Section 5.2 were repeated on AVHRR scenes and MIDAS air temperature measurements from May and September and for all summer months between 06:00 and 08:00. The new data were added to the existing database table to provide a total of 2120 paired EST and air temperatures records.

To generate a predictive model of air temperature from EST, linear regression was used to fit a line of best fit to all paired EST and air temperatures. This would allow an estimation of air temperature to be derived from EST based on the EST-air temperature relationship captured over the 23 year time-series (1985-2008) by the available AVHRR and MIDAS data from the four London weather stations. A least-squares first order polynomial fit function was used to generate the regression. A least-squares approach was chosen as it is simple to apply (Ebdon, 1985), but is more rigorous than other regression approaches such as the semi-averages method (Hammond and McCullagh, 1978). Furthermore, a linear fit was chosen in-line with the literature (Vogt et al., 1997; Fung et al., 2009; Nichol et al., 2009) and as visual inspection of the data indicated a linear relationship (e.g. see Figure 5.10). The equation of the fitted-line would then be used to derive estimates of air temperature from

EST.

A Python module, building on the software framework described in Chapter 4 was created to undertake linear regression and modelling. The satellite derived air temperature module (SDAT, see Appendix I) can perform linear regression on EST and air temperature for model calculation using data in the uniform NumPy array data model (see Section 4.1.2). Listing 5.1 shows the `buildmodel` function from the SDAT module (see Appendix I for the complete module source code). The build model function accepts EST and air temperature arrays and uses the NumPy polyfit function (Ascher et al., 1999) to compute a least squares first order (linear) fit between EST (the independent variable) and air temperature (the dependant variable). The function returns the equation of the fitted line in the form of slope and y-intercept. Using the `buildmodel` function (Listing 5.1) the line equation for the EST and air temperatures could be calculated and used to form a predictive model for air temperature. The implementation of the line-equation to estimate air temperatures is discussed below in Section 5.3.5.

```
def buildmodel(surface_temperature, air_temperature):                    20
    '''Creates model based on linear relationship between surface and
       air temperatures, returns slope and y-intercept.'''
                                                                         22
    x = surface_temperature
    y = air_temperature                                                  24

    z = np.polyfit(x,y,1)                                                26

    return z[0], z[1]                                                    28
```

Listing 5.1: An extract from the SDAT module showing the buildmodel function, returning the line equation of a least-squares first order fit between EST and air temperature (see Appendix I).

5.3.3 Cross-validation

To evaluate the quality of air temperature prediction using the relationship as defined by the linear regression of EST and air temperatures, cross-validation was used (Geisser, 1993). Cross-validation allows testing of predictive relationships derived from a correlation between two data-sets without the need for additional test data (Geisser, 1993). As such, cross-validation was selected for use in this study as all available pre-processed AVHRR and MIDAS data were used to define the EST-air temperature relationship, meaning that no additional data were available for validation. Cross-validation works by removing a sub-set of the original data (the validation sample) and using the remaining data points (the construction sample) to derive a new line of best fit to act as the model. The new model is then used to predict values of the dependant variable. The newly predicted values are then compared to the validation sample to assess the quality of prediction (Geisser, 1993).

To test EST and air temperatures four different cross-validation tests were performed, each generating root mean square error (RMSE) as the test metric to quantify the resulting errors. RMSE was selected as it is a popular method for succinctly quantifying the error of residuals (Ebdon, 1985). Each of the four cross-validation tests were designed to assess the response of the model to different characteristics. The first test performed a standard two-fold cross-validation, which was used as a benchmark for the other cross-validation tests. In the two-fold test the paired EST and air temperatures were randomly divided into two with the first half being used as construction data for the model and the second half as validation data (Geisser, 1993). The process is then repeated with a second pass, but with the construction and validation data swapped, and an average of the RMSE values from both passes calculated. Figure 5.6 shows the first pass of the two-fold cross-validation process. A two-fold cross-validation function was created as part of the SDAT module and can be seen in Appendix I.

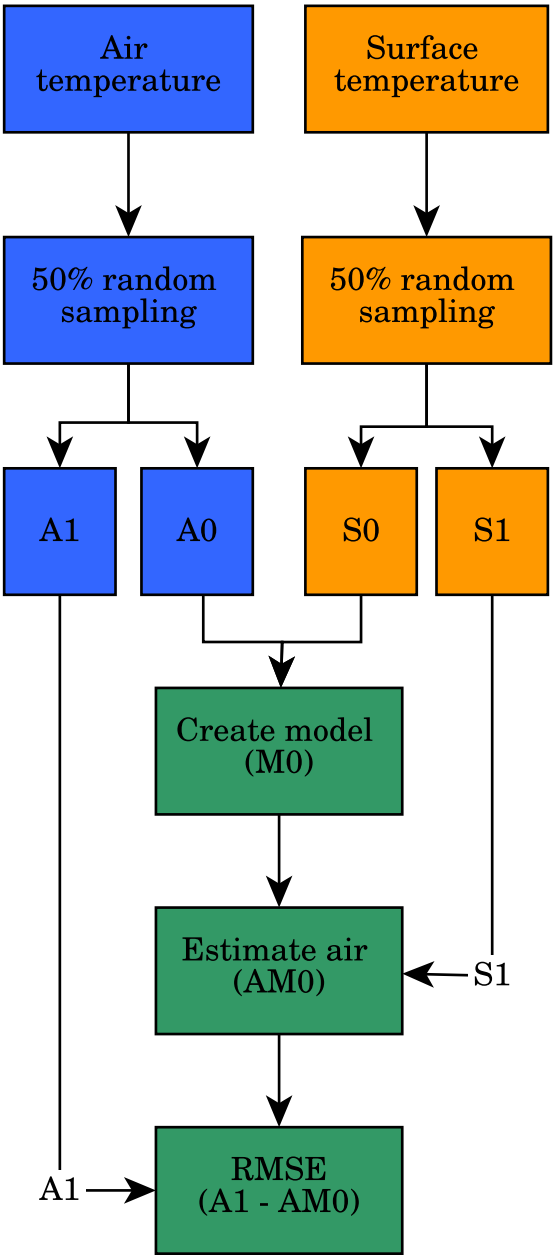


Figure 5.6: Flow diagram showing the two-fold cross-validation (first pass) testing of the air temperature model.

The second cross-validation test was a leave-one-out test where a single observation is removed from the paired data, and the remaining observations are used to construct a new line of best fit. The removed observation is then used for validation, generating an RMSE value. This process is iterated over all the data and average of all RMSE values forms the test statistic. The advantage of the leave-one-out test is that all observations are used for both training

and validation. An RMSE higher than the two-fold test would indicate an uneven weighting in the data, with some points having a considerably greater effect on the fit than others. The leave-one-out cross-validation test was also implemented as part of the SDAT module and is included in Appendix I.

The third test was similar to the two-fold test, except that the data removed from the model to act as validation data were manually selected by station. All the data from each of the four stations were removed in turn to act as the validation sample and the remaining three stations used for the construction of the model. The objective of this test was to quantify the model's dependency on each of the stations. A large RMSE for a given station would indicate that the model was highly dependant on the relationship between EST and air temperatures at the removed station, and that the three stations used to build the model were unable to predict air temperature at the removed station.

The fourth test paired EST and air temperature measurements from six summers in the time-series to act as validation samples (1989, 1990, 1995, 2002, 2003, 2006). The six summers were selected from the time series because they had high numbers of paired measurements ($n > 100$) and so could be used to form robust validation samples. Each summer was tested in turn, with the rest of the time-series used for model construction. The results were compared to the maximum, average and minimum EST and air temperatures for each of the six tested summers to identify relationships between cross-validation results and different summer temperature regimes (e.g, cool, average and warm summers). The objective of this test was to assess whether the derived models could predict air temperatures for the removed summers. This test would examine the accuracy of the model to estimate air temperatures during the extreme heatwave summer of 2003. Given the extreme temperatures recorded during the summer of 2003 (Burt, 2004a) it is reasonable to predict that these outlying temperatures will be less well characterised by the model than mean temperatures during the time-series. Therefore, the results from this test provide a key indicator as to whether the model is capable of estimating air temperatures during heatwave events. The third and fourth cross-validation tests were created as Python scripts which used the cross-validation functions

in the SDAT module (Appendix I), for brevity these scripts have not been included.

5.3.4 Testing for systematic error

A possible systematic error within the model is the difference between the observation time of EST and air measurements. As was previously described in Section 5.2.3 the time difference between paired EST and air temperatures could be up to 30 minutes. Due to the nature of the satellite data requiring cloud free days and the study focusing of summer temperatures the periods used in the study represent clear, warm days when solar gain is maximised during the morning and heat loss via thermal radiation into the atmosphere is maximised in the evening (Oke, 1987).

As a result, the days used in the study are likely to have experienced rapid warming and cooling throughout the day. During a rapidly warming or cooling period an error could be introduced where there is significant change in either EST or air temperature between the time of EST and air temperature measurement. For example, during the morning of a particular day between 06:00 and 14:00 assuming a linear increase in temperatures of 10°C, then a satellite-based EST measurements at 11:30 could be up to 0.63°C warmer than an EST temperature at 11:00 leading to a potential systematic error between EST and air temperature at 11:00. For this reason, prior to the use of the model to create London-wide estimates of air temperatures the paired EST and air temperature measurements were examined for significant differences between measurements recorded at the same time (i.e. on the hour) and those with larger time differences of up to 30 minutes. If a systematic error was detected then it may be possible to model the error as a function of time difference between EST and air temperature measurements and apply a correction to temperatures to improve the correlation between the two, and in-turn improve the derived linear regression.

Measurements were divided into morning (06:00 to 14:00) and afternoon (14:00 to 21:00) groups. Morning observations were assumed to exhibit warming,

whilst afternoon observations were assumed to exhibit cooling, as is commonly found in urban areas (Oke, 1987). Observations were grouped into three categories based on the difference in time between EST and air temperatures measurements: "early" observations where the satellite over-pass was 15-30 minutes before air temperature observation, "late" observations where the satellite over-pass was 15-30 minutes after air temperature observation and "on-time" observations where satellite overpass was less than $\pm$15 minutes before/after air temperature measurement. The data were grouped into early and late observations to distinguish between increases and decreases in temperature between EST and air temperature measurement time. For example a morning "early" observation (assuming a linear temperature rise) would lead to a cooler EST measurement than EST taken at the time of air temperature measurement, whereas an afternoon "early" observation (assuming linear temperature decrease) would lead to a warmer EST measurement than one taken at time of air temperature measurement. The threshold of 15 minutes was chosen as EST within $\pm$15 minutes showed no visual difference when plotted against EST measurements taken exactly at the time of air temperature observation.

Morning and afternoon, "early", "on-time" and "late" paired EST and air temperatures were plotted with trend lines. Analysis of the plots showed that differences between the different groups was minimal (see Section 5.4.4) and that there were an insufficient number of measurements in the afternoon groups for statistical testing. A Mann-Whitney U test at the 95% confidence level was used to test for significant difference between "early" and "late" morning measurements. Given the small differences seen between data in different groups, "early" and "late" measurements were tested against each other as they should exhibit the greatest difference in temperature, with "early" morning measurements being cooler than "late" measurements, and therefore most likely to exhibit a statistical difference.

5.3.5 Generating model outputs

The intercept and slope coefficients generated from the linear regression model between EST and air temperature using the `buildmodel` function were used to create an SDAT function `model` to estimate air temperature from input surface temperature (see Appendix I). The `model` function calculates an estimate of air temperature by multiplying an input EST value plus the y-intercept by the slope value. Additionally the `model` function uses the uniform data model (NumPy arrays), so that spatially-complete estimates of air temperature could be generated from input AVHRR EST scenes. Hard-coding the intercept and slope coefficients into the SDAT module removes the requirement to enter them manually at run-time or call the `buildmodel` function to return them, saving processing time. An additional function `runmodel` was created, which accepts intercept and slope coefficients and EST value, and returns estimated air temperature, for use in cases where the original line values aren't required such as during the cross-validation testing (see Appendix I).

To evaluate the spatial robustness of the model a test similar to the station cross-validation (Section 5.3.3) was employed. The objective of this test was to understand the dependency of the model on each of the stations to predict air temperatures over the entire GLA area, and whether the model was still valid when derived with fewer than four stations. This is important for future studies, to understand whether four stations are suitable for deriving such relationships. Four new models were created, each based on the line of best fit between EST and air temperature measurements where records from one of the four stations had been removed. Air temperature for the GLA was then predicted using each of the four models, based on input EST values which were a temporal average of all summer daytime AVHRR 1.1Km EST from 1985 to 2008 (the same dataset used to generated Figure 5.13). Comparisons were then made between each of the four predicted air temperature grids, and predicted air temperature from the model derived from all four stations described above (Figure 5.13).

To evaluate the model's ability to represent individual summers, six further

grids of estimated air temperature were produced using the original model, with input ESTs calculated from temporal averages of AVHRR ESTs from each of the six summers identified during cross-validation (1989, 1990, 1995, 2002, 2003 and 2006; see Section 5.3.3). Comparison of the resulting predicted air temperatures against the air temperatures measured by the MIDAS weather stations for the six selected summers were used to see how the model predictions responded to different input surface temperatures. The estimates from the summer of 2003 would reveal whether the model can estimate air temperatures based on EST values from a heatwave summer, and as such provides an indicator, along with aforementioned summer cross-validation (see Section 5.3.3), of the utility of the model to derive spatially-complete estimates of air temperature during extreme temperature events.

5.4 Results

5.4.1 The relationship between EST and air temperature

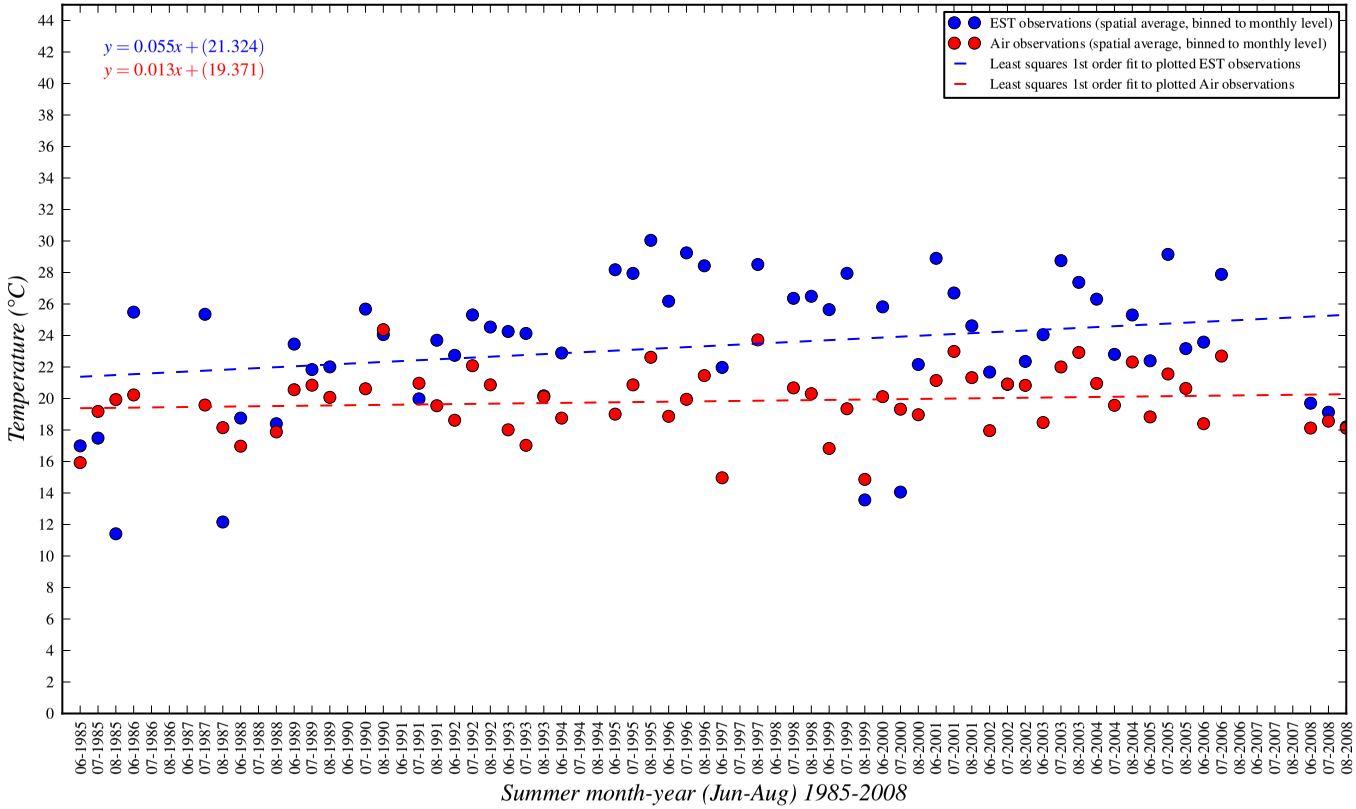
Figure 5.7 shows the spatial averages of EST and air temperature measurements, temporally averaged to the monthly level, over the time-series. The plot shows that both EST and air temperature data exhibit similar fluctuations over the time-series. Table 5.4 shows the summary statistics for both of the time-series. On average over the time-series EST is 3.54°C warmer than air temperature, and is more variable, with EST standard deviation ($\sigma_{EST} = 4.32°C$) being twice that of air temperature ($\sigma_{Air} = 1.98°C$). This is reflected in the maximum and minimum EST values which are 5.67°C and -3.45°C warmer and cooler than average air temperature respectively, broadly agreeing with the relationships seen in the literature (e.g. Nichol et al. (2009)). Such results are to be expected as air temperature is less conductive to kinetic and radiative heat transfer (air is often considered a thermal insulator) than the urban surface, meaning that EST has a greater range than air temperatures in the urban boundary layer (Oke, 1987). The highest ESTs over the plot appear to coincide with known heatwave events (e.g.

August 1995, July 2003), although this pattern is somewhat less discernible with the air temperatures. There is also a difference between the first order linear trend in EST and air temperatures. Whilst air temperature appears to exhibit a minimal gradient (0.013), the EST shows a steeper gradient (0.055), although the cause of the apparent divergence between the two is unclear.

	$\overline{x}$ (°C)	σ (°C)	**Min** (°C)	**Max** (°C)
EST	23.38	4.32	11.41	30.04
Air	19.84	1.98	14.86	24.37
Δ	3.54	2.34	-3.45	5.67

Table 5.4: Summary of averaged EST and air temperatures over the time series.

The Wilcoxon test (see Section 5.3.1) between the spatial averages of EST and air temperature from the four stations resulted in rejection of the null hypothesis in favour of statistical difference between EST and air temperatures at the 95% significance level. The results from the Wilcoxon test for difference between EST and air temperatures at each of the four stations (Section 5.3.1) are shown in Table 5.5. These results show a split between stations in acceptance and rejection of the null hypothesis. The stations SJP and NTH accepted the null hypothesis and have EST and air temperatures that can be considered to be statistically similar. In contrast the stations LWC and LHR have p-values considerably lower than the rejection level and therefore EST and air temperatures can be considered statistically different at these stations. These results indicate that station location controls the relationship between EST and air temperature, particularly as SJP and NTH stations are in urban areas with high vegetation cover and previous studies have shown that a correlation between vegetation density and relationship between EST and air temperature exists (Prihodko and Goward, 1997). Furthermore, as LHR and LWC represent locations of extreme urban environments, where turbulent air flow is more likely (Oke, 1987), it is less likely that EST and air temperatures will be well correlated (Nichol et al., 2009).



Number of scenes per month/year for EST values

Month	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
Jun	1	0	4	3	15	12	10	1	1	0	12	5	2	8	3	4	6	7	23	4	9	0	0	3
Jul	11	0	3	0	5	20	6	1	2	1	6	7	0	1	3	2	4	9	11	2	5	19	0	8
Aug	4	9	0	3	12	0	0	4	6	8	11	6	3	0	2	4	5	17	9	1	9	13	0	4

Figure 5.7: Monthly averaged summer daytime EST and air temperatures and trends for selected stations in GLA

Station	Significance Level	P-Value	Null Hypothesis
LWC	0.05	3.66×10^{-7}	Rejected
SJP	0.05	0.36	Accepted
LHR	0.05	5.18×10^{-5}	Rejected
NTH	0.05	0.25	Accepted

Table 5.5: Results from the Wilcoxon signed rank test for difference between paired EST and air temperatures at different London weather stations. *Stations; LWC: London Weather Centre, SJP: St James's Park, LHR: London Heathrow, NTH: Northolt.*

The results from the Mann-Whitney U-test between stations for EST and air temperature (Section 5.3.1) are shown in Tables 5.6 (for air temperature) and Table 5.7 (for EST). Table 5.6 shows that for air temperature the null hypothesis can only be rejected for one of the station combinations (LWC-NTH) at the 95% significance level. For the remaining station combinations air temperature measurements were found not to be significantly different. The results of this test suggest that overall air temperature is not spatially variable and that for the duration of the time series there is little difference in air temperature between stations, although the LWC-NTH result indicates that the strength of the relationship may vary.

Station	LWC	SJP	LHR	NTH
LWC	-			
SJP	0.05211 (accept)	-		
LHR	0.1001 (accept)	0.5793 (accept)	-	
NTH	0.03697 (reject)	0.9893 (accept)	0.5609 (accept)	-

Table 5.6: P-values and null hypothesis status from the Mann-Whitney test for significant difference between hourly air temperatures at the selected London weather stations. *Stations; LWC: London Weather Centre, SJP: St James's Park, LHR: London Heathrow, NTH: Northolt.*

In contrast, Table 5.7 shows that EST appears to exhibit spatial variability, with four of the six station combinations rejecting the null hypothesis and exhibiting significant difference at the 95% significance level. However, the EST results still show ambiguity with station pairs LWC-LHR and NTH-SJP not showing any significant differences. Interestingly, this pairing of stations

matches the results from the Wilcoxon testing between EST and air temperatures (Table 5.5), suggesting that the pairs of stations have similar relationships between EST and air temperatures.

Station	**LWC**	**SJP**	**LHR**	**NTH**
LWC	-			
SJP	0.009 (Reject)	-		
LHR	0.67 (Accept)	0.009 (Reject)	-	
NTH	0.007 (Reject)	0.70(Accept)	0.0034 (Reject)	-

Table 5.7: P-values and null hypothesis status from the Mann-Whitney test for significant difference between AVHRR scenes of EST at different London weather stations over the available time series. *Stations; LWC: London Weather Centre, SJP: St James's Park, LHR: London Heathrow, NTH: Northolt.*

The preceding tests showed that EST and air temperatures which were spatial averages of values from the four weather stations are significantly different at the 95% significance level. However, at the station level only two of the four stations exhibited statistical equality between EST and air temperatures at the 95% significance level (SJP and NTH). These results suggest that the relationship between EST and air temperatures is dependant on location, and as such the use of spatial averages from the four stations are potentially not suitable to derive a correlation between the two. Furthermore, examining the relationships between the four stations in EST appears to confirm this, as the station pairs which were statistically equal have corresponding acceptance/rejection of the Wilcoxon test for a relationship between EST and air temperatures. However, when examining inter-station differences in air temperature the pattern is less clear with five of the six station pairs exhibiting statistical equality at the 95% significance level, indicating that EST captures more significant inter-station differences than air temperature.

Figure 5.8 shows a scatter plot with the correlation between spatial averages of EST and air temperatures. Figure 5.9 shows a scatter plot and correlation between EST and air temperatures from all four stations. Each plot also contains a first-order line of best fit and an r^2 value derived from Pearson's correlation coefficient. A comparison between the plots shows that the spa-

tially averaged EST and air temperatures have a lower level of correlation, resulting in a much lower r^2 value (0.26) as compared to the individual measurements (0.51). The decreased r^2 value from the spatially-averaged data in conjunction with the above statistical testing show that spatially-averaging EST and air temperatures over the four stations drastically reduces the level of correlation between them. These results support the hypothesis that the EST-air temperature relationship is location dependent, and as such relationships between the two should be derived with spatially-paired data.

Furthermore, the plots also show confidence limits at the 95% interval for predictions of air temperature. The confidence intervals denote the most probable area within which a predicted value is likely to fall if it was estimated using an EST not included in the original regression. Both plots show narrow confidence limits, indicating that predicted air temperatures (with 95% confidence) will have a small variation from the regression line. However, the second plot (Figure 5.9) shows a reduction in prediction variability of cooler temperatures as a result of using the individual measurements, with a narrowing of the confidence limits between EST of 5°C and 10°C as compared to the correlation from spatial averages of EST and air temperature (Figure 5.8).

The tests in this section have shown that the relationship between EST and air temperatures varies with station location, and are statistically similar at two of the stations (SJP and NTH), but are not at LHR and LWC. The ambiguity in these results indicates that stations SJP and NTH are more conducive to forming a closer relationship between EST and air temperatures than LHR and LWC, potentially as a function of urban land cover and land use (Oke, 1987; Nichol, 1994; Prihodko and Goward, 1997; Nichol et al., 2009). The use of individual measurements as opposed to the spatial averages of EST and air temperature from the four stations shows a better level of correlation. However, EST only explains up to 52% of the variation in air temperature, suggesting that summer daytime EST and air temperatures are not that well correlated.

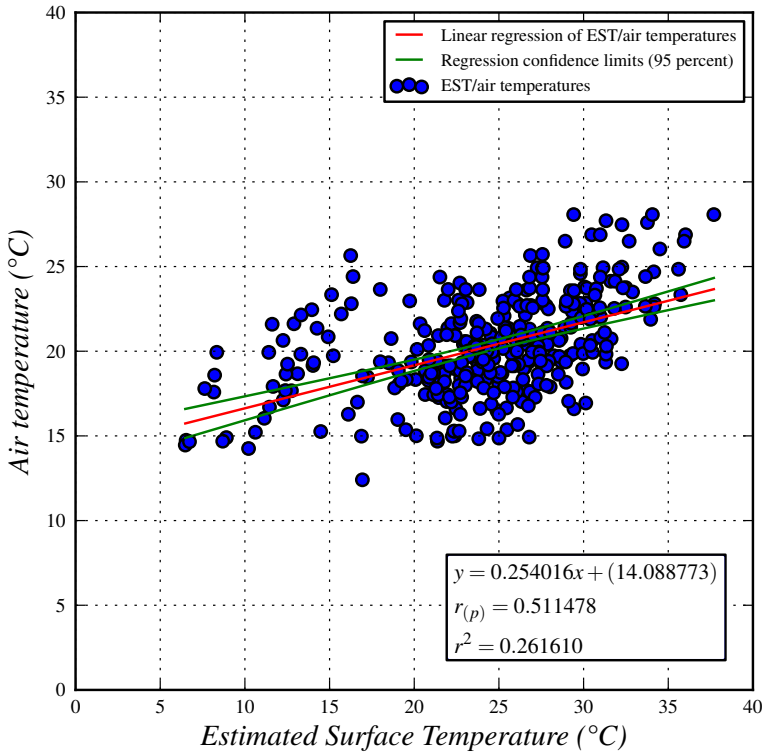


Figure 5.8: Scatter plot of spatial averages of EST and air temperatures from all four weather stations (08:00-21:00, June-August) 1985-2008.

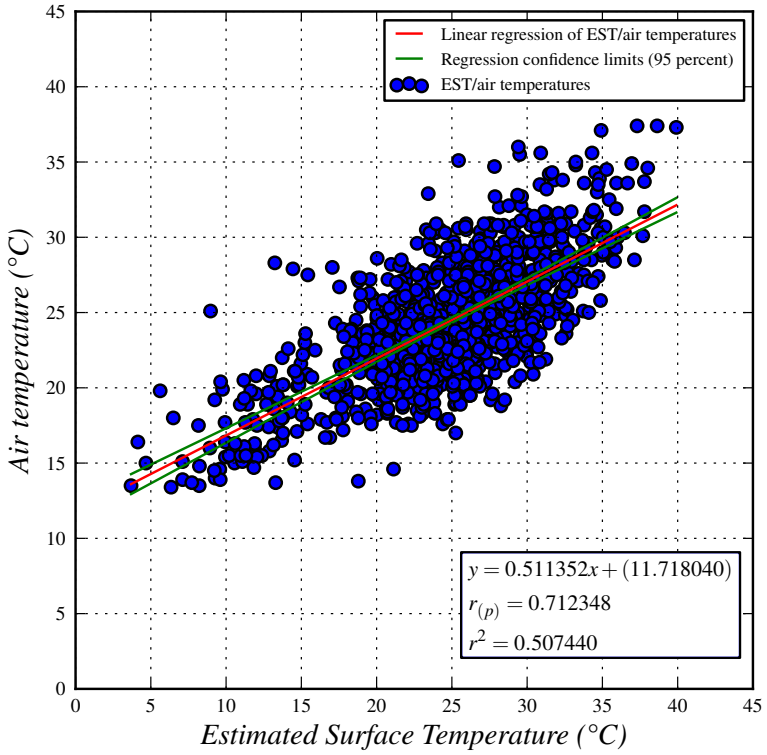


Figure 5.9: Scatter plot of individual EST and air temperatures over stations (08:00-21:00, June-August) 1985-2008.

5.4.2 Modelling air temperature

As described in Section 5.3.2 data from May and September for times between 06:00 and 08:00 was added to the existing set of paired measurements to improve the diurnal and monthly temporal coverage of the EST and air temperature data. Figure 5.10 shows the relationship between EST and air temperature with the new data as compared to the original data in Figure 5.9. A comparison between the two shows very little change in the slope of the line of best fit between the old and new data-sets, suggesting that the addition of the new data hasn't changed the nature of the relationships examined in the previous section (Section 5.3.1). However, it can be seen that the new data from May and September as well as between 06:00 and 08:00 contains a larger number of cooler temperatures than the original data-set. This is to be expected of measurements from the cooler summer months (May and September) and earlier in the diurnal cycle. As a result the fit at the lower end of the plot, where previously there were fewer measurements, has been improved, leading to an increase in the r-squared value between EST and air temperature (r^2 = 0.68) and a narrowing of the confidence limits. Using the linear regression derived between EST and air temperatures including the new data the equation for the line of best fit was calculated: $T_{air} = 0.571404 \times T_{surface} + 9.918152$. This equation was subsequently used to form the predictive model of air temperature.

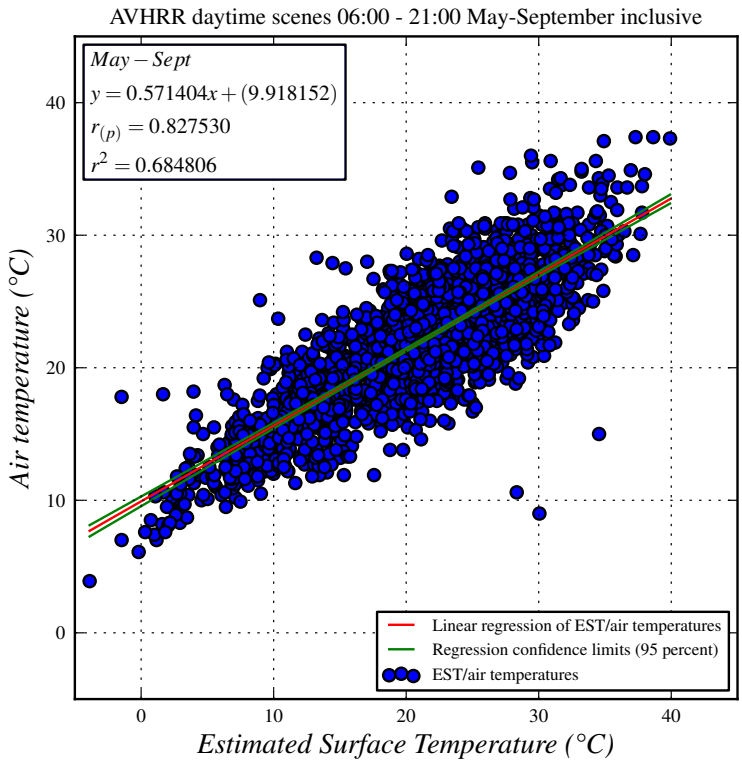


Figure 5.10: Scatter plot of individual EST and air temperatures with extended data period (06:00-21:00, May-September) 1985-2008.

5.4.3 Cross-validation results

Results from the two-fold and leave-one-out test are shown in Table 5.8. The two-fold test shows an RMSE value of 5.03, indicating that the model has a reasonable accuracy in prediction of air temperature, although this is greater than variations between neighbourhood temperatures in the literature which have shown to have significant impact on the variability of heat hazard exposure (e.g. <5°C, Harlan et al. (2006)). As previously described the two-fold cross-validation value was used as the cross-validation benchmark by which to evaluate the response of the data to other cross-validation tests.

The leave-one-out cross-validation test shows a minor decrease in RMSE of 0.01. The minimal difference between the results of the two tests suggests that observations are evenly weighted, and that during two-fold cross-validation when 50% of the data are removed at random the remaining construction sample is still able to predict values with the same magnitude of error as the

leave-one-out test which removes observations individually, one at a time.

Table 5.9 shows the results of the cross-validation tests when individual stations are removed. In contrast to the two-fold and leave-one-out tests all the RMSE values from station removal are up to 2.33 lower than the two-fold test. The lower RMSE results can be attributed to these tests removing a smaller number of validation samples, leaving a greater number number to act as the construction sample. For example, in the two-fold test 50% of the data are randomly removed (1062 measurements) while for each of the station tests between ~17-31% of all data were removed (371-672 measurements). This meant that for each station test a much larger construction sample was available, giving a more robust line of best fit.

In Table 5.9 it is also apparent that the RMSE values for each of the station tests have very small differences between them (maximum RMSE variation of 0.24). This demonstrates that despite the inter-station variability in the relationship between air and surface temperatures as described in Section 5.3.1, the stations all have an approximately equal capability within the model. This means that for each of the four stations removed to act as the validation sample, data from the other three stations are capable of estimating temperatures at the validation station with the same magnitude of error (0.45-1.67°C) that is lower than both the two-fold and leave-one-out cross-validation tests. The small variation in RMSE values between stations appears to correspond with the number of measurements at each station. LHR and LWC which both have the greatest number of observations (therefore reducing the remaining number of observations available to act as a construction sample resulting in a model with a poorer fit) have the highest RMSE values (2.92 and 2.94 respectively), suggesting that the model may be more representative of temperatures recorded at these two stations than NTH and SJP, which could lead to over-prediction of temperatures in urban green-space such as that found at SJP.

Table 5.10 shows the results of cross-validation for different summers. The highest RMSE (3.11) occurs during removal of observations from 2006 which

had an average air temperature of 24.31°C. The lowest RMSE (2.51) is found in 1995 and 2002 with average air temperatures of 25.67°C and 19.64°C respectively. These results show that there is no apparent relationship between the RMSE and mean average air temperature for each summer, even during summers like 1995 and 2003 which contained known heatwave events (Koppe et al., 2004; Johnson et al., 2005; Kovats et al., 2006). It would be reasonable to assume that model prediction of summers containing heatwaves would be worse than non-heatwave summers, as extreme events would only account for a small number of measurements in the original data, furthest away from the line of best fit. However this test has shown that the model is capable of predicting temperatures for heatwave summers with the same magnitude of error as other non-extreme event summers.

Cross-validation Method	**Construction Sample**	**Validation Sample**	***RMSE***
Two-fold	1062	1062	5.03
Leave-one-out	2124	1 (iterated over all)	5.04

Table 5.8: Results of the two-fold and leave-one-out cross-validation tests for the EST and air temperatures (May-September 06:00-21:00).

Station Removed	**Validation Sample**	***RMSE***	$\bar{x}_{air}$ (°C)	σ_{air} (°C)	$\bar{x}_{surface}$ (°C)	$\sigma_{surface}$ (°C)	$\bar{x}_{air} - \bar{x}_{surface}$ (°C)
LWC	662	2.94	21.44	4.95	20.99	7.54	0.45
SJP	371	2.70	22.97	4.76	20.81	6.32	1.55
LHR	672	2.92	21.72	5.26	21.02	7.94	0.69
NTH	419	2.79	22.32	4.98	20.65	6.57	1.67

Table 5.9: Results of station cross-validation tests for EST and air temperatures (May-September 06:00-21:00). Standard deviation and mean temperatures are for the station removed. *Stations; LWC: London Weather Centre, SJP: St James's Park, LHR: London Heathrow, NTH: Northolt.*

Year Removed	**Validation Sample**	***RMSE***	$\bar{x}_{air}$ $(°C)$	σ_{air} $(°C)$	$\bar{x}_{surface}$ $(°C)$	$\sigma_{surface}$ $(°C)$	$\bar{x}_{air} - \bar{x}_{surface}$ $(°C)$
1989	113	2.75	20.68	4.32	19.66	7.14	-1.02
1990	120	2.73	21.7	5.83	18.98	8.7	-2.72
1995	102	2.51	25.67	3.41	27.9	4.68	2.23
2002	176	2.51	19.64	5.22	16.95	6.74	-2.69
2003	271	2.94	23.45	5.33	23.95	6.7	0.14
2006	179	3.11	24.31	4.24	24.2	5.32	-0.11

Table 5.10: Results of summer cross-validation analysis for the surface and air temperatures (May-September 06:00-21:00). Standard deviation and mean temperatures are for the year removed. *Stations; LWC: London Weather Centre, SJP: St James's Park, LHR: London Heathrow, NTH: Northolt.*

5.4.4 Testing for systematic error

Figure 5.11 shows morning paired air and surface temperature measurements, grouped by early, late and on-time satellite overpass times. If the assumptions stated in the methodology (Section 5.3.4) are correct then air temperatures paired with early over-passes would be warmer than those with on-time over-passes. In contrast, air temperatures paired with late over-passes would be cooler than those paired with on-time over-passes. Therefore, the differences between early/late and on-time trend-lines in Figure 5.11 could indicate a systematic error introduced by the change in temperature between observations with a time-delta of $\pm$15 minutes or greater.

The difference between the trend lines in Figure 5.11 (although minimal $\leq$2.0°C) correspond with the hypothesis of temperature change between the times of the two measurements. The line of best fit for paired measurements where AVHRR EST was captured 15-30 minutes after air temperature, indicates lower air temperatures in these pairings, as compared to EST-air temperature measurements within 15 minutes of each other. This could be the result of surface warming (leading to higher EST) in the 15-30 minute interval after the air temperature measurement. Furthermore, the trend line for paired measurements where EST was captured 15-30 minutes before air temperature, show warmer air temperatures as compared to those measurements

captured with 15 minutes of each other. As such, this could also indicate warming in the 15-30 minute interval before air temperature measurement, leading to a cooler EST being paired with a warmer air temperature.

Figure 5.12 shows a plot for afternoon observations. However, these data exhibit no clear distinction between early or late groups. It is likely that this is as a result of too few measurements in the afternoon data, when evaporation is greatest, leading to a reduced number of AVHRR scenes available due to cloud contamination. Given that the afternoon plot showed no possible systematic error, afternoon observations were not investigated for a temporal offset between satellite over-pass and air temperature measurement.

The results for the Mann-Whitney U test between early and late morning temperatures (see Section 5.3.4) found no significant difference between the two. This shows that the slight variation seen in the trend lines is not statistically significant at the 95% confidence level. The trend lines show an apparent difference between early and late measurements, possibly as a result of ± 30 minutes between temporally-paired EST and air temperatures, however the results from the statistical testing indicate that this doesn't cause a significant systematic error between early and late measurement groups, and is therefore unlikely to affect the validity of the regression model.

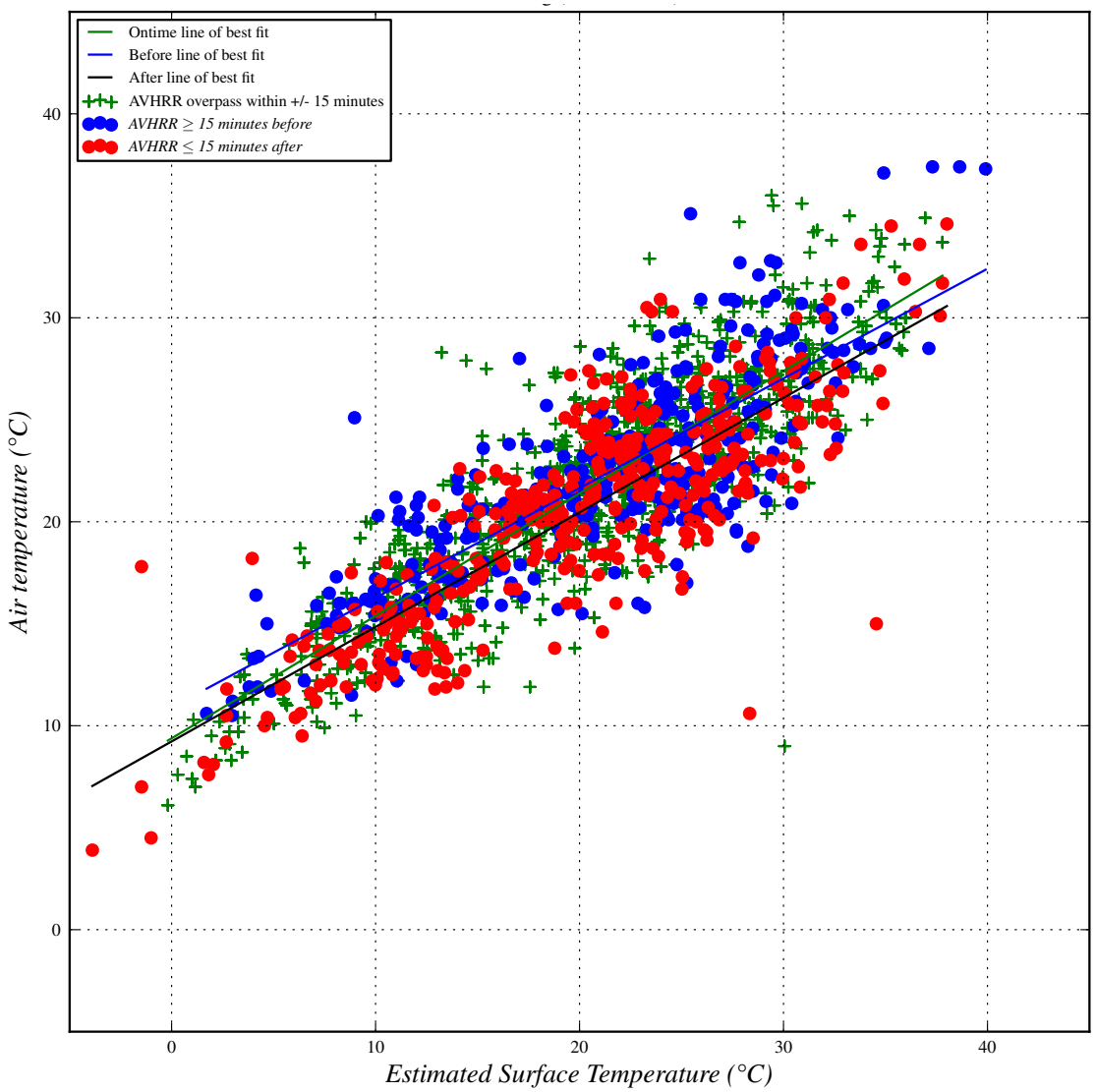


Figure 5.11: Scatter plot of morning (before 14:00) early, late and on-time paired EST and air temperatures.

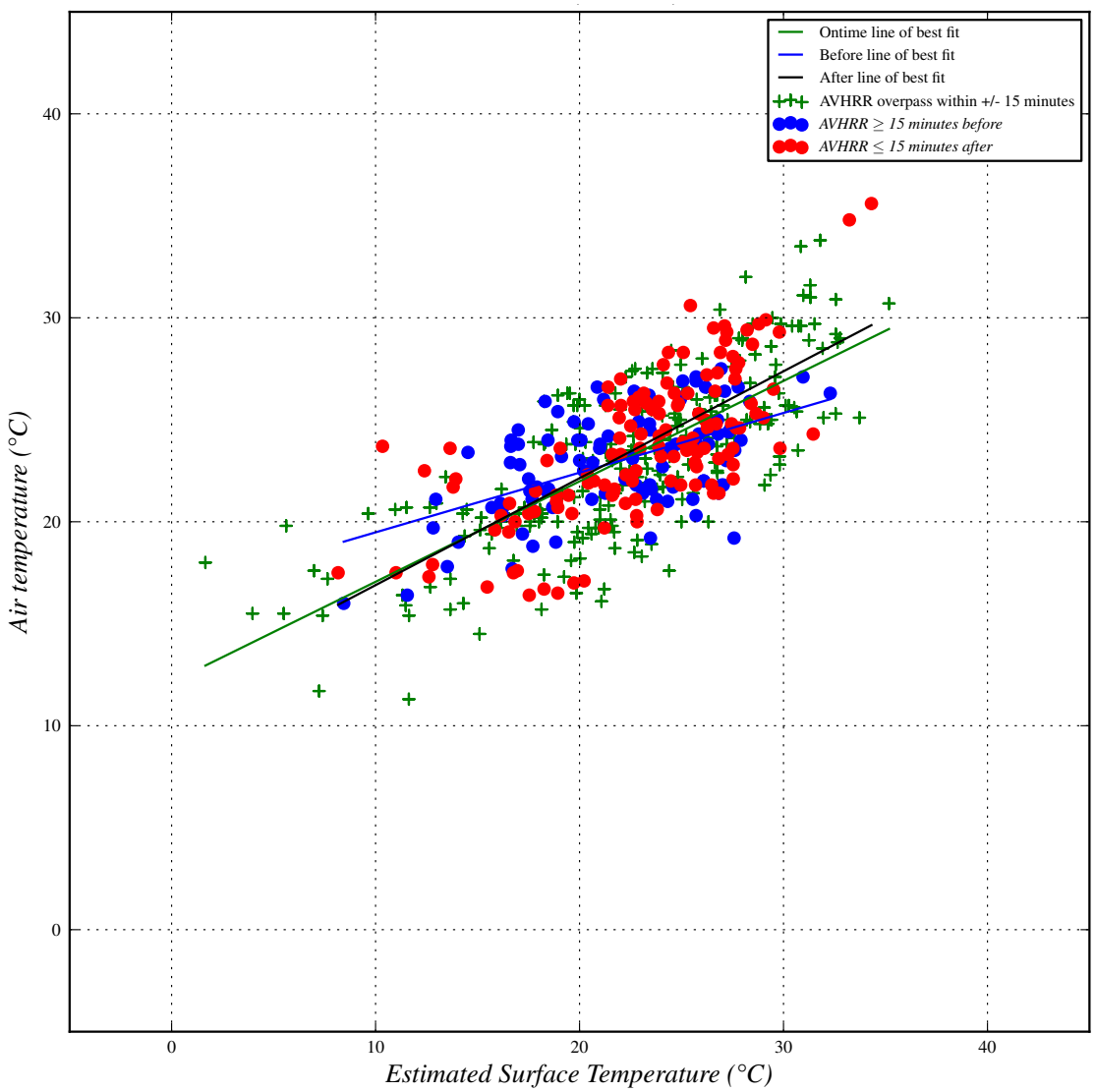


Figure 5.12: Scatter plot of afternoon (after 14:00) early, late and on-time paired EST and air temperatures.

5.5 Estimated air temperatures for Greater London

Figure 5.13 shows modelled air temperatures for the GLA (b) derived from input EST from all available daytime scenes, temporally averaged over the time-series from 1985 to 2008 between May and September (a). Table 5.11 shows the summary statistics of the temporally averaged ESTs and the estimated air temperatures. From Figure 5.13 it is clear that the modelled air temperatures exhibit a similar spatial pattern of temperature to the ESTs,

with for example, cooler regions such as the River Thames and urban-rural fringe being clearly visible. The estimated air temperatures exhibit a lower overall variation in temperatures (standard deviation of estimated air temperatures is 0.78°C lower than EST, Table 5.11). This is to be expected somewhat given that air temperature tends to have lower variation that EST (Nichol et al., 2009).

Compared to air temperatures from the four weather stations (an average of the air temperatures paired with AVHRR ESTs, over the time-series), the estimated air temperature shows large differences between the minimum and maximum air temperatures from the time-series. This is somewhat to be expected as the model air temperatures were generated using ESTs averaged over the time-series and so are not representative predictions of the coolest and hottest temperatures recorded between 1985 and 2008. This is also reflected in the differences between standard deviations of estimated and measured air temperatures ($\Delta\sigma$=-4.0). However, the difference between averages of measured and modelled air temperatures is more promising, with a difference of 1.30°C, suggesting that the model is capable of estimating average temperatures over the time-series with relatively small differences between averages of predicted and measured air temperatures.

Data	**Min (°C)**	**Max (°C)**	$\overline{x}$ **(°C)**	σ **(°C)**
AVHRR EST (series average)	18.29	26.67	23.17	1.81
Modelled air temperature	20.38	25.16	23.16	1.03
Measured air temperature (series average from air temperatures paired with EST)	3.9	37.4	21.86	5.03
Δ Measured-Modelled	16.4	-12.24	1.30	-4.00

Table 5.11: Statistics for EST (average over time-series), modelled air temperature and measured air temperatures (average of air temperatures from the four stations over the time-series).

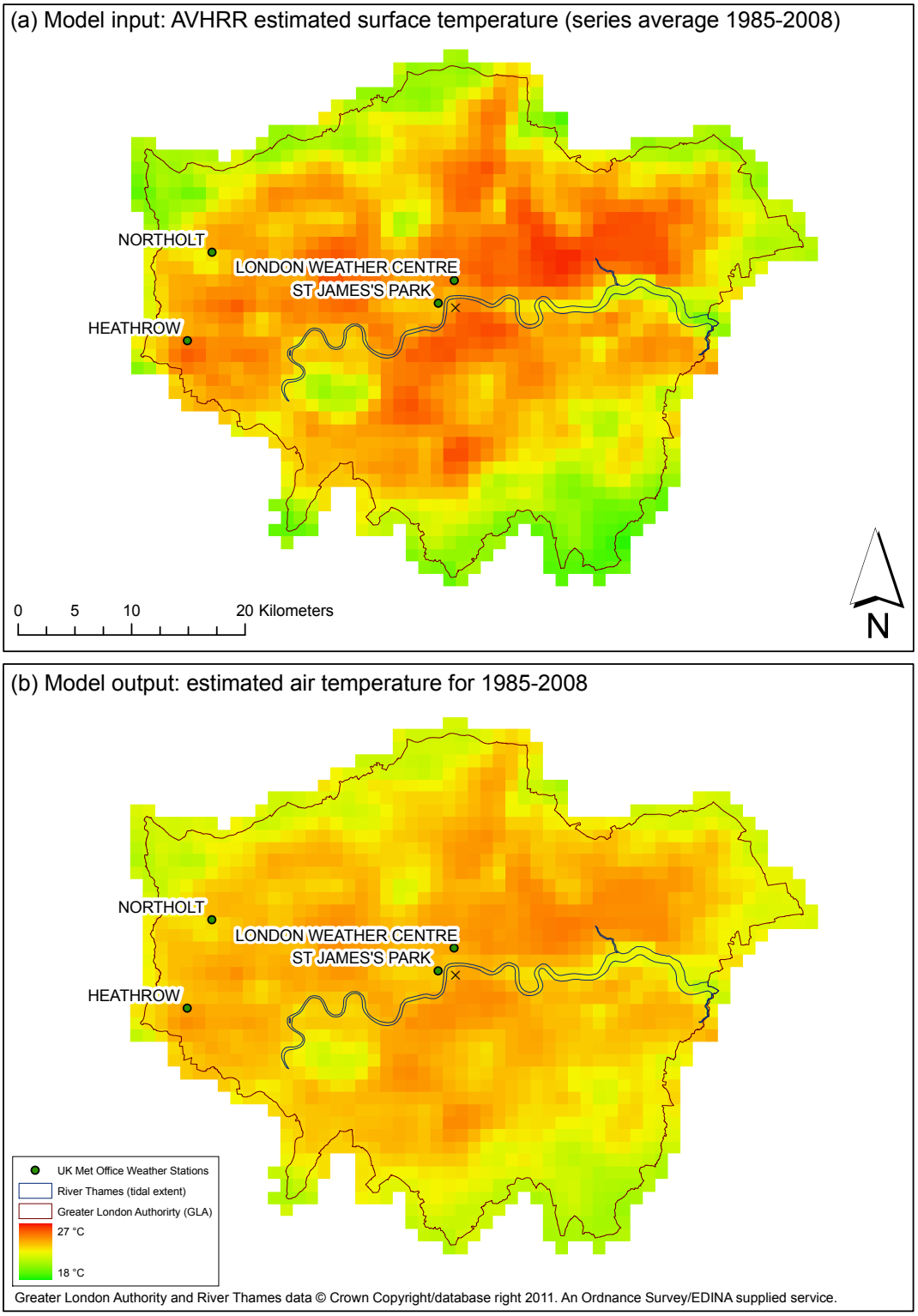


Figure 5.13: EST and estimated air temperatures over the Greater London Authority for daytime (06:00-21:00) summer months (May-September) between 1985-2008.

Figure 5.14 shows modelled air temperatures as derived from the relationship between EST and air temperatures, with paired measurements from

each of the four weather stations removed in turn (see Section 5.3.2). Table 5.12 shows the summary statistics from each of the output air temperature estimates. These results show that there is very little variation in the spatial pattern of predicted air temperature between the different grids. The summary statistics show only small differences in the average temperatures between the derived grids, with a maximum difference in the mean and standard deviation of 0.35°C and 0.07 respectively (Table 5.11). Furthermore, all four scenes of estimated air temperature (Figure 5.14) appear slightly warmer than air temperatures predicted using the model derived from data of all four stations (Figure 5.13(b)), with a maximum mean difference of 0.62°C (Table 5.12). These results show that the empirical model derived from just three stations can predict air temperature with variation less than 1°C as compared to the model derived using data from all four stations, supporting the results from the cross-validation testing of Stations (5.4.3) that each triplet combination of the four stations is capable of predicting air temperatures without a prominent change in the error of estimation.

	Min (°C)	**Max (°C)**	$\overline{x}$ **(°C)**	σ **(°C)**
All stations	20.38	25.16	23.16	1.03
LWC	21.21	25.62	23.78	0.95
SJP	20.99	25.01	23.45	0.98
LHR	20.98	25.50	23.61	0.98
NTH	20.98	25.18	23.43	0.91

Table 5.12: Statistics for estimated air temperatures using models derived from the EST-air temperature relationship with each weather station removed. *Stations; LWC: London Weather Centre, SJP: St James's Park, LHR: London Heathrow, NTH: Northolt.*

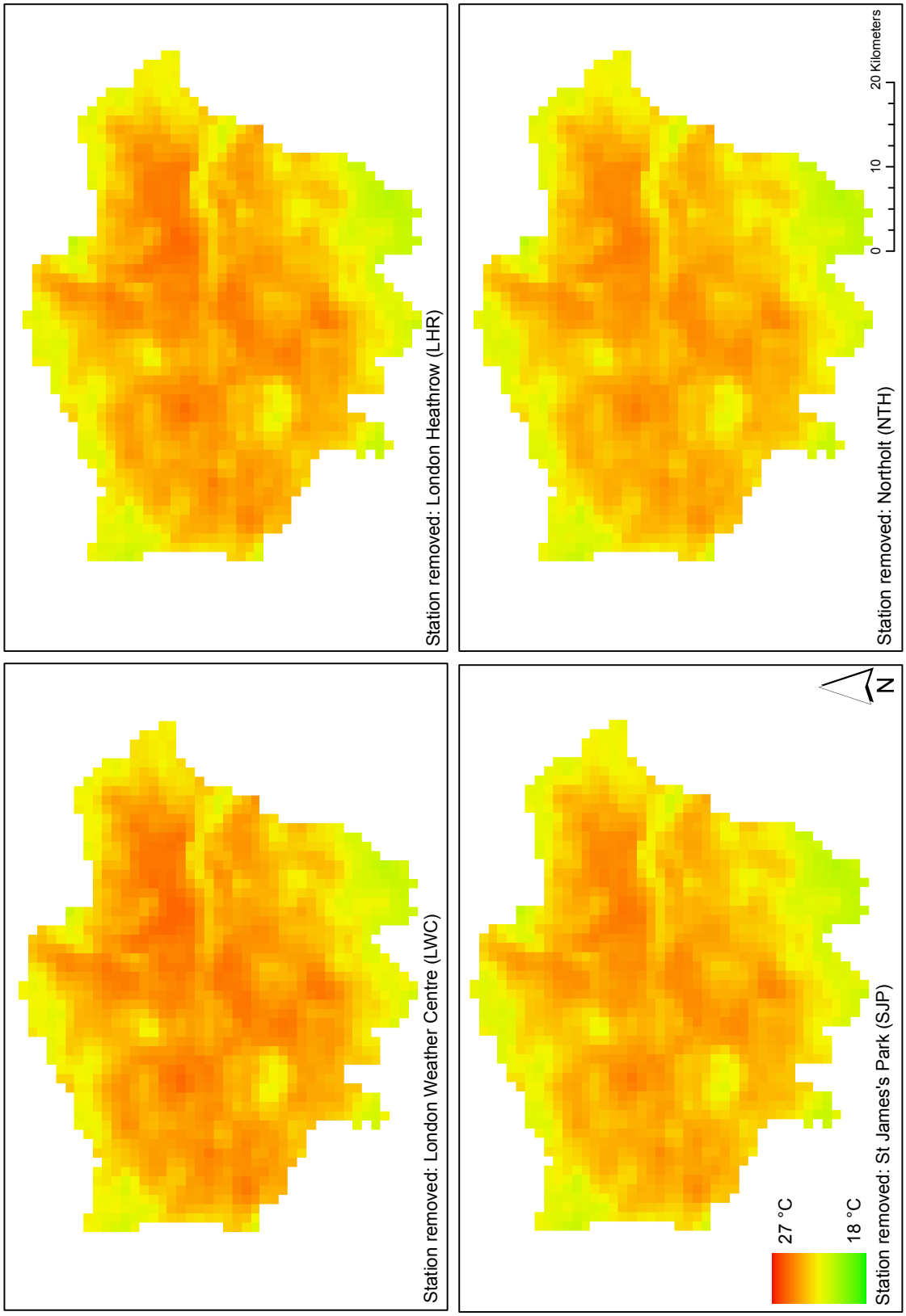


Figure 5.14: Modelled air temperature with different stations removed from the model.

Figure 5.15 shows the results of modelled air temperatures for different summers (see Section 5.3.5), and Table 5.13 shows the summary statistics for the modelled air temperatures as well as the air temperature measurements from the specified summer. The table shows that generally, predicted air temperatures are similar to observed air temperatures at the four stations, with differences between measured and predicted temperatures for all years under 1.5°C. The largest difference between the predicted and measured air temperatures is 1.44°C, found during the summer of 1990. The summer with the lowest difference was 1995 (0.23°C). Furthermore, the variations in predictions between summers correspond with the input surface temperature, for example EST from the known heatwave summers of 1995, 2003 and 2006 produce the highest predicted air temperatures. These results show that the model is capable of predicting air temperature values representative of input summer EST, including summers which contain extreme temperature events.

Due to the range of values in the six summers, the temperatures of the heatwave summer of 1995 (average air temperature from four stations: 25.44°C) appear exaggerated in Figure 5.15. Whilst this could be the result of AVHRR sample size for each year (see Section 6.4.2), in this case it is unlikely as for the summers under consideration the number of AVHRR scenes used to create the temporal aggregates of EST for each summer (used as model inputs) ranged from 22 to 33 (see Figure 5.7).

Year	**Measured Air Temperature (averaged over stations)**		**Predicted Air Temperature (averaged over GLA)**		$\bar{x}_{measured} - \bar{x}_{predicted}$ $(°C)$
	$\bar{x}$ $(°C)$	σ $(°C)$	$\bar{x}$ $(°C)$	$\sigma(°C)$	
1989	20.68	4.32	20.44	0.80	0.24
1990	21.7	5.83	20.26	0.68	1.44
1995	25.67	3.41	25.44	1.01	0.23
2002	19.64	5.22	19.09	0.89	0.55
2003	23.45	5.33	22.94	0.90	0.51
2006	24.31	4.24	23.14	1.10	1.17

Table 5.13: Statistics for estimated summer air temperatures (using ESTs from the six selected summers) and corresponding air temperature measurements.

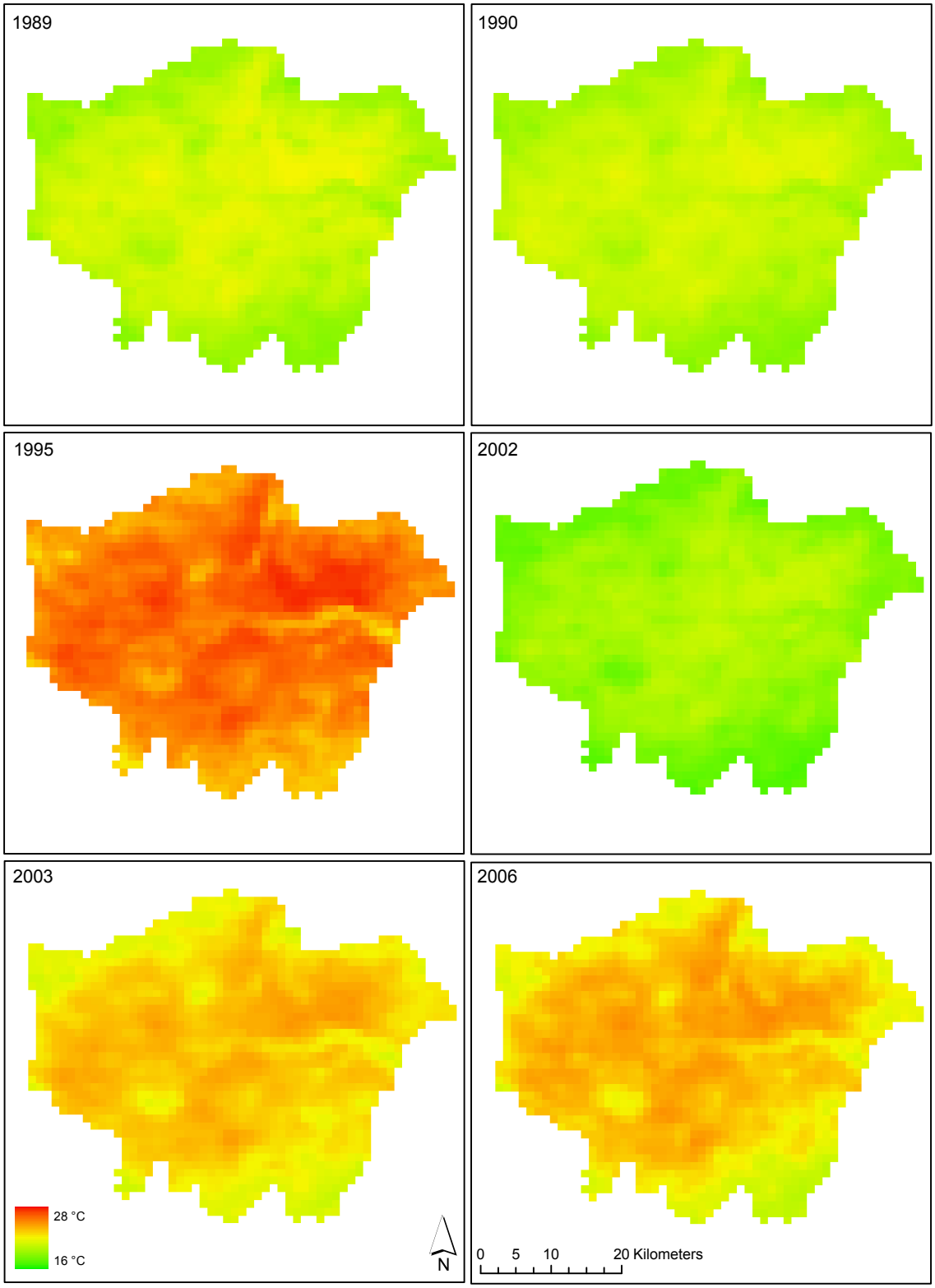


Figure 5.15: Estimated summer air temperatures using ESTs from the six selected summers.

5.6 Summary

The analysis presented in this chapter has demonstrated the ability to predict summer daytime near-surface air temperature from Earth observation estimated surface temperatures using a predictive empirical model. The long-temporal baseline of data in conjunction with temporal and spatial pairing of EST and air temperatures meant that given a suitable number of measurements a high-level of correlation (r^2= 0.68) could be established without further processing. Such a relationship means that future analysis of urban temperatures using long temporal baseline ESTs can produce a spatially complete estimate of air temperature, if required to replace or supplement in-situ terrestrial measurements. This data could be used to quantify intra-urban spatial variability in increases of exposure to the heat hazard during heatwave events, for heatwave mitigation and adaptation studies (Harlan et al., 2006; Dousset et al., 2011).

The first objective of the work was to quantify the relationship between air and surface temperatures, especially with regard to spatial variability. The statistical testing between stations was only partially conclusive, with two of the four stations exhibiting significant differences in EST and none exhibiting significant differences in air temperature. However, spatial averages of EST and air temperature measurements showed a reduction in correlation of 0.26 (r^2), validating the requirement for spatial pairing of individual measurements for such analysis.

The second objective was to evaluate whether the observed relationship between EST and air temperatures could be used to form an empirical predictive model of air temperature. Cross-validation testing was used for this purpose and showed RMSE values of 5.03 and 5.04 for two-fold and leave-one-out tests respectively. Cross-validation of different stations showed spatial robustness in the model, with all four station combinations exhibiting RMSE values up to 2.09 lower than the benchmark two-fold cross-validation. This is important with respect to future research, as it suggests that given a long time-series of data four stations can be used to derive a spatially robust model. Additionally,

cross-validation testing using six selected summers as the validation sample showed RMSE for estimates of air temperature in the selected summers at least 1.92°C lower than the two-fold cross-validation test. This is a key finding as it suggests that the model is able to predict temperatures from summers experiencing a variety of temperatures (e.g. heatwave, average, cool) with reasonable accuracy, even if the summer is not included in the original model. Potentially, this means that the model could be used to characterise air temperatures for future heatwave events not in the time-series, although further work is required to test this hypothesis.

Furthermore, prior to testing model outputs, paired morning and afternoon measurements were examined for systematic bias as a result of differences between AVHRR and weather station observation time. These results showed no significant difference between early and late morning measurements, proving that temporal pairing within ± 30 minutes is suitable for defining empirical relationships between air and satellite Earth observed EST. No relationship was discernible between the early and late afternoon measurements, likely the result of too few afternoon measurements being available due to cloud cover.

The model was analysed by employing a station-based test, similar to the cross-validation method, where four new models were generated, each with one station removed, and the results of which were compared to the complete model. These results showed that an empirically derived model using just three stations could predict air temperature with less than 1°C difference from a model derived using four stations. This confirms the station cross-validation results, showing that using four stations provides spatial robustness in predictions.

Lastly, the complete model was used to generated estimated air temperatures using EST for six specific summers. The six generated estimated air temperature images, based on temporal averages of EST for the specific summers, showed that the model was capable of predicting air temperature values for the heatwave summer of 2003 as well as the non-heatwave summers with

similar levels of accuracy (maximum average error 1.44°C, Table 5.13).

The above results validate the use of long-time series, spatio-temporally paired satellite EST and terrestrial air temperature observations for empirically based urban air temperature prediction to examine extreme temperature events in a spatially-continous and complete manner. However while the reported results are encouraging there is scope for further research, particularly with regards to investigating the EST-air temperature relationship further to create better predictions of individual summers during extreme events and annual temperature variations within urban environments. This could include comparison of measurements within the diurnal cycle and using night-time scenes, though additional sources of air temperature data would also be required to better validate the predicted air temperatures.

Chapter 6

Evaluating the response of London's UHI to summer temperature changes using AVHRR data

6.1 Introduction

The urban heat island effect increases the exposure of urban populations to the heat hazard during heatwave events (Johnson et al., 2005; Kovats and Hajat, 2008; Rosenzweig et al., 2011). Over the course of the 2003 heatwave summer the increase in heatwave mortality of Londoners was 25% greater than the rest of England and Wales (Table 2.6) (Johnson et al., 2005), which was attributed to the urban heat island and increased pollution levels in the city (Johnson et al., 2005). Furthermore, previous studies have found that the urban heat island is spatially variable (Harlan et al., 2006; Kolokotroni and Giridharan, 2008). For example Kolokotroni and Giridharan (2008) found an average difference of 1.3°C between the urban centre and semi-urban areas of London during clear and calm conditions over one year. It has been recognised that it is critical to quantify the urban heat island in a spatially explicit

manner in order to identify where the heat hazard is greatest, so that mitigation and adaptation options for future heatwaves can be targeted to areas with the greatest heat stress (Harlan et al., 2006; Dousset et al., 2011).

Traditionally, the urban heat island has been quantified using the urban heat island intensity metric (Equation 2.1), derived using screen-level air temperature measurements from terrestrial weather station networks. However, as discussed in Section 2.3.1, the utility of the UHII metric derived from terrestrial weather station data to quantify spatially intra-urban temperature dynamics is often limited by the number of urban weather stations available (Kawashima et al., 2000; Vogt et al., 1997).

Thermal remote sensing has been widely used to derive spatially-complete EST to act as an alternative or supplementary data source for weather station measurements to quantify intra-urban temperatures (Roth et al., 1989; Nichol, 1996; Tomlinson et al., 2012). As such, a number of studies have used EST to derive the UHII metric (Pongracz et al., 2006; Cheval and Dumitrescu, 2009; Cheval et al., 2009; Tomlinson et al., 2012) using the difference between urban and rural pixels to generate spatially-complete UHII across different cities (Pongracz et al., 2006; Tomlinson et al., 2012). However, previous studies have highlighted the importance of having sufficient time series of remotely sensed data, which is often limited due to sensor operational characteristics (Tran et al., 2006). A suitable time-series is required to be able not only to derive metrics that are insensitive to local diurnal fluctuations, such as meteorological conditions, but also capture the long-term spatial and temporal temperature dynamics required for monitoring (Tran et al., 2006). In this regard, it is worth noting that spatial and temporal sampling and time-series length have also been recognised as a critical consideration in UHII studies using terrestrial weather station screen-level air temperature data (Jones and Lister, 2009).

In relation to the last point, the long time series AVHRR EST data archive developed in this project (see Section 3.2) could potentially be used to create spatially and temporally robust measurements of temperature for the subse-

quent calculation and analysis of the UHII. The high revisit times from the large number of satellites carrying the AVHRR sensor (see Section 3.2), result in up to four scenes a day being acquired (Cracknell, 1997). This offers the potential to characterise temperature dynamics at a daily, weekly, monthly and seasonal time scale. While a positive aspect of the sensor, it is important to recognise that understanding temperature dynamics, particularly in relation to annual changes and extreme events inter-annually means that data from different years need to be compared. This raises the question as to what the ideal temporal frequency of AVHRR data is in order to capture consistently and objectively annual changes in temperature and in particular extreme events such as heatwaves.

With respect to the issues described above, the research presented in this chapter investigates the utility of the AVHRR data to capture, characterise, and quantify the magnitude of summer season temperatures in order to evaluate whether such data can be used to distinguish a heatwave summer from non-heatwave years. In this regard, an evaluation of the commonly used UHII metric generated from both terrestrial MIDAS air temperature measurements and the daytime AVHRR scenes is undertaken to quantify the metric's ability to capture a known heatwave event. The methodology for the research was split into two objectives (Figure 6.1). First, an assessment of the temporal sensitivity of EST derived from AVHRR data was conducted to identify the ideal temporal frequency of AVHRR data to employ in order to consistently and objectively capture the changes in temperature caused by a heatwave event, rather than other erroneous factors such as local meteorological conditions (Figure 6.1(b)). Second, using the results of temporal sensitivity assessment, the UHII metric was derived using AVHRR EST values to capture spatially the magnitude of London's urban heat island during heatwave and non-heatwave summers. UHII was also derived in the traditional manner using the MIDAS air temperature measurements for the same time periods as the AVHRR data. Using both these data-sets an evaluation of the UHII metric to capture significant changes in intensity during a heatwave was undertaken (Figure 6.1(c)).

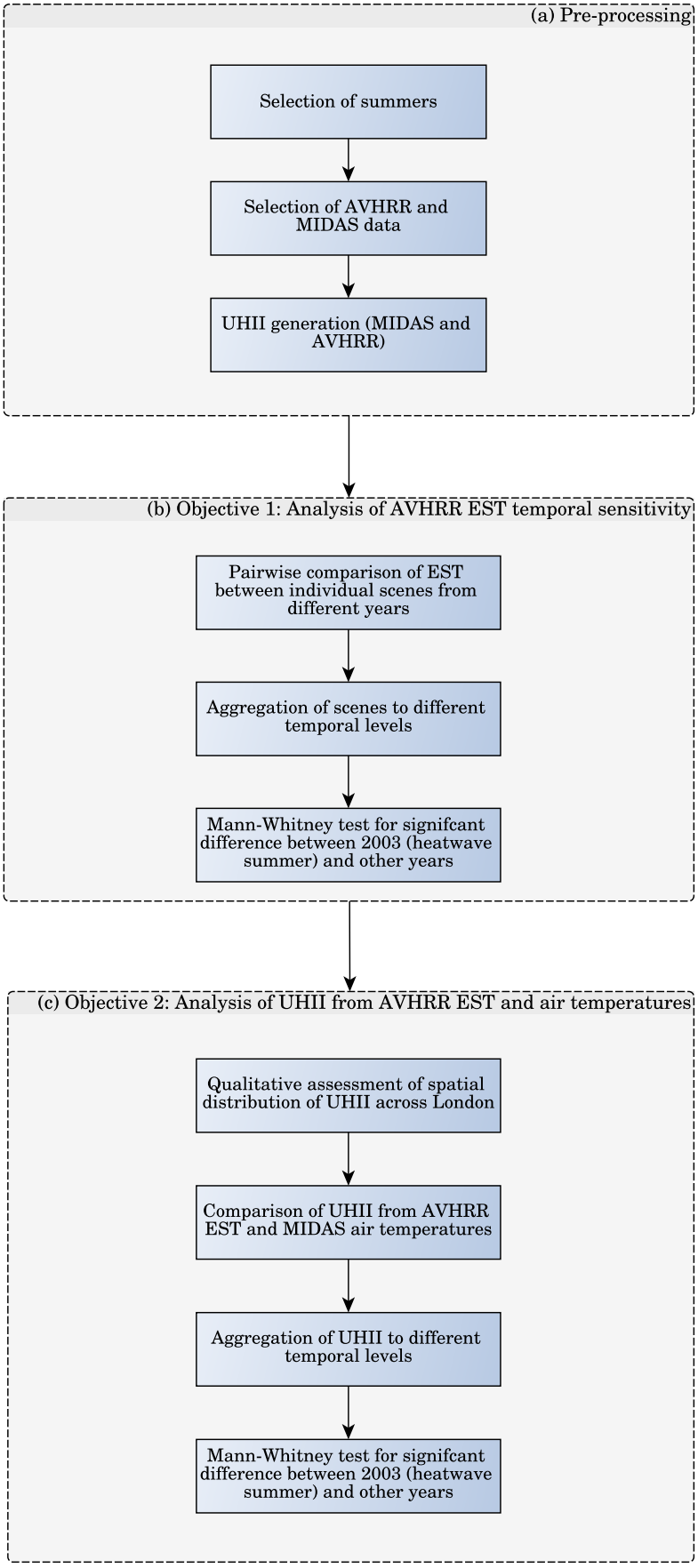


Figure 6.1: Flow-line showing the methodology for pre-processing and analysis of AVHRR temporal sensitivity and the UHII using both EST and air temperatures.

6.2 Data selection and pre-processing

6.2.1 Selection of MIDAS and AVHRR data

In order to perform analysis of MIDAS screen-level air temperatures and AVHRR ESTs during a known heatwave summer and compare these to temperatures during non-heatwave summers, a sub-selection of the AVHRR EST and MIDAS air temperature data archive (see Sections 3.1 and 3.2) was created. This was achieved using a 11 year period from 1996 to 2006, which was selected as it includes the heatwave summer of 2003, one of the hottest heatwaves on record in Europe (Burt, 2004a) and which resulted in a severe increase of mortality in London (see Section 2.3.2). To select summers for analysis, average summer temperatures (1st June to 31st August 1996-2006) were generated from the four London weather stations (Section 5.2.1; London Weather Centre (LWC), London Heathrow (LHR), Northolt (NTH) and St James's Park (SJP)) using their corresponding daily (full diurnal cycle) averages in the PostGIS database (Table 6.1).

Year	**Average summer air temperature (°C)**
1996	17.62
1997	18.24
1998	16.78
1999	18.12
2000	17.24
2001	17.93
2002	17.45
2003	19.36
2004	18.01
2005	17.90
2006	19.28

Table 6.1: Average summer (1st June -31st August) air temperatures for 1996-2006 from diurnal averages of four London weather stations (LWC, LHR, NTH and SJP).

The 11 summers were ranked according to average summer daily temperature and MIDAS hourly air temperatures (full diurnal cycle) and AVHRR EST daytime scenes for the hottest, 75th, 50th and 25th percentile summers were

selected for further analysis (Figure 6.1(a)). Selected summers and their corresponding average screen-level air temperatures for all four London weather stations were; hottest 2003 (19.36°C), 75th percentile 1997 (18.24°C), median 2001 (17.93°C), and the 25th percentile 2002 (17.45°C). The four summers were chosen in this manner to keep the number of AVHRR scenes to process (i.e. to generate UHII from EST) viable, while ensuring that the general temperature record of London for the period was captured. For the chosen summers there were a total of 81 AVHRR EST scenes available for the Greater London Authority; 43, 5, 14 and 19 scenes for 2003, 1997, 2001 and 2002 respectively.

In addition to the four urban stations (see Section 5.2.1), the weather station High Wycombe (22 kilometres from the GLA boundary) was included in the MIDAS hourly air temperature data for analysis to act as a rural station for generation of the UHII metric (Figure 6.2). The High Wycombe (HWC) station was selected in a similar manner to the four urban stations (Section 5.2.1), due to its rural nature and good quality hourly data record over the required time series (94.4% completeness of diurnal hourly measurements between 1996 and 2006).

The location of the High Wycombe station was also used in the generation of AVHRR EST UHII (Figure 6.1(a), Section 6.2.2). However, as previously discussed in Section 3.2.8, prior to EST generation the AVHRR data was subjected to additional cloud screening. The cloud screening showed that the rural station at High Wycombe had a higher rate of cloud contaminated pixels than those within the GLA boundary (see Section 3.2.8). As a result of cloud contamination over the rural station, only 75 of the 81 AVHRR EST scenes could be used for generation of the UHII metric from AVHRR EST data. For the four years under consideration 2003, 1997, 2001 and 2002 there were 43, 3, 12 and 17 scenes available cloud free over the GLA and rural weather station for UHII generation.

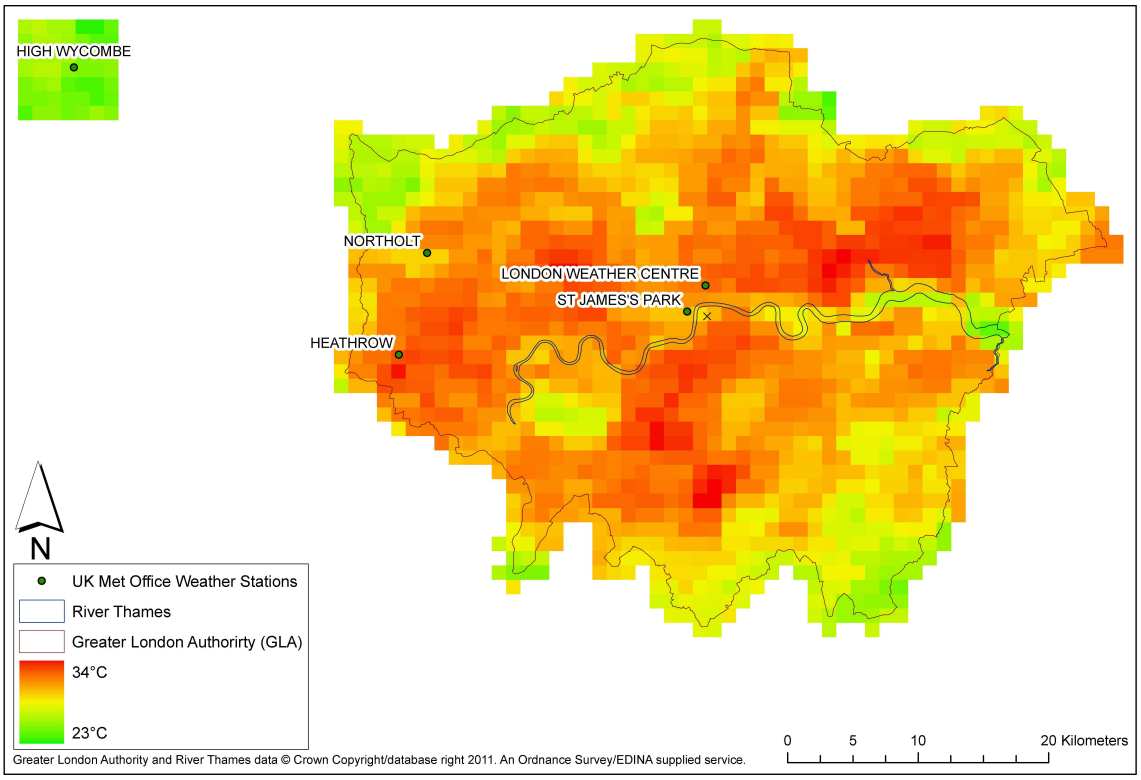


Figure 6.2: AVHRR scene of Greater London processed to estimated surface temperature (EST), 8th August 2003, 14:04 (GMT). Also shown are the location of the London weather stations employed in the study including the rural reference site relative to London.

6.2.2 Generating the urban heat island intensity metric

For the purposes of this study, the UHII metric is formally defined as the maximum difference between urban and background rural temperatures for a given point in time during one diurnal cycle (see Section 2.3.1, Equation 2.1, Oke (1987); Kim and Baik (2002); Kolokotroni and Giridharan (2008)).

Screen-level air temperature UHII was generated from the MIDAS PostGIS database using an SQL script to calculate the hourly difference between the rural station High Wycombe and each of the four urban stations for each day in the selected summers. The urban-rural station pair with the greatest difference in temperature for each diurnal cycle was then selected as the UHII for that day. Four new tables were created in the database containing the UHII values for each of the selected years. Each UHII record in the tables included the corresponding urban station and the hour within which the max-

imum difference was found.

Table 6.2 shows an extract of the database table containing UHII values for 1997. In the table the fields **urb_pid** (urban primary ID) and **rur_pid** (rural primary ID) are the unique ID's for the hourly measurements from the specified urban and rural weather stations used to generate the UHII value. These values can act as foreign keys to the observation primary key recorded in the hourly air temperature table (see Section 3.1.3), so that each UHII value can be traced back to its component urban and rural air temperatures if required. In the extract from 1997 (Table 6.2) it can be seen that London Heathrow (LHR) had the greatest difference in temperature compared to the rural station for the first three days in June 1997, all of which occurred between 22:00 and 23:00. Further details of the UHII results are described in detail in Section 6.4.3 below.

urb_stn	rur_stn	urb_pid	rur_pid	ob_date	ob_time	uhii
LHR	HWC	22443591	22443573	1997-06-01	22:00	3.9
LHR	HWC	22450500	22450482	1997-06-02	23:00	4.8
LHR	HWC	22456922	22456904	1997-06-03	22:00	5.1
SJP	HWC	22457416	22457345	1997-06-04	00:00	4.8

Table 6.2: An extract of the MIDAS screen-level air temperature UHII table for the summer of 1997 (1st June - 31st August). *Note: urb_stn and rur_stn codes replaced by station letter codes for clarity.*

To calculate daytime UHII using the AVHRR ESTs, a slightly different approach had to be taken due to the reduced observation frequency of the AVHRR data compared to the terrestrial weather station data. At best, a maximum of two daytime scenes were available per diurnal cycle. Therefore, the UHII metric was calculated using individual AVHRR EST scenes. This was done by subtracting the EST of the pixel containing the rural station High Wycombe from the EST values of the pixels within the Greater London Authority boundary in a manner similar to that used by Tomlinson et al. (2012) to derive UHII measurements from MODIS EST scenes. This process was achieved using the PyRaster software in the processing framework (Section 4.2). A Python function "`rasterUHII`" in the PyAVHRR module was created to subtract the value of the pixel containing the rural station from the rest of the remaining pixels

in the scene. The `rasterUHII` function uses the `getpixelval` function developed as part of the the database interface (see Section 4.3, Listing 4.10) to get the EST value of the rural pixel, which is then subtracted from all other values in the NumPy pixel array. Using the `rasterUHII` function EST UHII was generated for each of the 75 AVHRR files from the selected summers.

6.3 Analysis methodology

6.3.1 Temporal sensitivity of AVHRR estimated surface temperatures

A two-step methodology was used to investigate the temporal sensitivity of AVHRR EST (6.1(b)). First, comparisons were made between scenes from 2003 and the other three years acquired within 65 minutes of each other on the same Julian day of the year (the additional five minutes was added to account for variation in the AVHRR orbit times, providing an additional three scenes for comparison). Differences in pixel values of EST between the scenes were created to see whether the scenes from the heatwave year (2003) were warmer than scenes taken at the same time (within 65 minutes) from the other three years. The comparisons were made using the software framework Psycopg database interface (see Section 4.3) to access the AVHRR meta-data database to select pairs of scenes which were captured within 65 minutes of each on the same Julian day in 2003 and other three years. Of the 81 EST scenes a total of 9 pairs within 65 minutes of each other (2003 and another year) were available for analysis. Each pair of scenes was loaded into NumPy arrays using the PyRaster module (Section 4.2) and the differences in temperature between pixels in both scenes was calculated and the minimum, maximum and average difference in EST between each pair of scenes was generated (Section 6.4.1, Table 6.3).

The second step was to generate temporal averages of all scenes for each of the four years. Temporal averaging has been investigated in marine sea surface

temperature studies using AVHRR data (e.g. Gentemann et al. (2003)). In such studies, scenes of sea surface temperature are averaged over one week with a weighting applied to night-time scenes in order to minimise diurnal signals (Gentemann et al., 2003). In order to investigate the sensitivity of urban temperature analysis and measurements of AVHRR EST and derived UHII values to temporal averaging, statistical comparisons were performed for two levels of temporal averaging between 2003 (the heatwave year) and the three other years under investigation.

Temporal averaging was performed at the monthly level (June, July, August) generating three scenes per year, and the summer level (1st June - 31st August) resulting in a single scene of EST per year. Temporal averaging at these levels was undertaken to be able to perform inter-year comparisons of monthly and seasonal temperatures. Weekly averaging was not performed as there were too few scenes to generate robust averages at the weekly level. Temporal averaging was achieved using the Raster Processing Suite Quantum GIS extension (Section 4.2.3). To generate temporal averages, all available scenes for the required time period were loaded into memory as NumPy arrays. Next, a single array of pixel averages was created by summing the values for each pixel in the input arrays and then dividing each pixel value by the number of scenes loaded, to create a mean average value for each pixel over the time period. The subsequent temporally averaged array was saved to disk as a raster image file.

For each level of temporal averaging the order of AVHRR ESTs was compared to the known ordering of the four percentile summers, to give a qualitative indication as to whether temporal averages of AVHRR ESTs represented the known temperature regime of each summer (for example whether the AVHRR EST for the summer of 2003 was hotter than temporal averages from the other summers in question). The non-parametric Mann-Whitney U-test of distributions was employed to test whether statistically significant differences existed between the 2003 (heatwave summer) scenes and the corresponding dates or averages (monthly and yearly) of the other three years under investigation (Figure 6.1(b)). This was applied such that each spatially coincident pixel

from a scene in 2003 was paired with its corresponding location (pixel) in either a monthly or summer scene from one of the other three years under investigation. In the statistical tests, a one-tail test was employed as the expectation was that the temperatures from 2003 should be greater than the other years. The null hypothesis was that there was no significant difference in the temporal averaged EST of 2003 compared to temporal averages from each of the other summers. The alternative hypothesis was that the extreme temperatures recorded during the heatwave summer of 2003 would cause EST values from 2003 to be significantly greater than those from the other three cooler summers. In all tests a 95% significance level was employed.

6.3.2 Analysis of urban heat island intensity metric

Based on the results from analysis of AVHRR EST temporal sensitivity (Section 6.4.1), temporal aggregates of UHII from AVHRR EST were created at monthly and seasonal levels for each of the four summers. In contrast, no temporal averaging was applied to the MIDAS air temperature UHII, as with only one value per day (Section 6.2.2) this would have provided too few average values to form a statistically robust sample. Furthermore, the objective of UHII testing with MIDAS air temperature data was not to assess the temporal sensitivity as with the AVHRR data (which is captured in discrete units of time, not continuously like hourly air temperature measurements). Instead, the objective was to support the analysis of the UHII metric, to quantify whether UHII values from terrestrial air temperature measurements captured a statistically significant increase in intensity during the heatwave summer of 2003 compared to the other summers.

Mann-Whitney U-tests were performed between the 2003 AVHRR EST UHII values and the other years investigated at monthly and yearly temporal averaging. A one-tail test based on 2003 having a intensity greater than other years was employed at a 95% significance level. The null hypothesis was that there was no significant difference in UHII values between 2003 as compared to the non-heatwave summers. The alternative hypothesis was that the ex-

treme temperatures recorded in the 2003 heatwave summer would result in significantly greater UHII values in 2003 as compared to the other three non-heatwave summers.

Mann-Whitney U-tests were also applied to the UHII values generated from MIDAS screen-level air temperatures. A one-tail test at the 95% confidence interval, based on 2003 having a greater intensity than the other years, was again used. The null hypothesis was that 2003 would not exhibit and significant increase in intensity. The alternative hypothesis was that the extreme air temperatures measured during the summer of 2003 would be reflected in a significant increase in intensity. Due to 25 days of missing data from the rural station over the four years (2003: 2, 1997: 12, 2001: 7, 2002: 4) the number of MIDAS air temperature UHII values per year was variable, and as such the Mann-Whitney U-critical value varied for each year tested. Therefore, the U-critical values for air temperature UHII of each summer are presented alongside U-calculated values in Table 6.9.

For comparison with air temperature UHII, the Mann-Whitney U-test was used to test for significant differences in average daily air temperatures from the four London stations between the heatwave summer 2003 and the other three summers. The air temperatures were tested using a one-tail test at the 95% significance level, based on 2003 being significantly hotter than the other years. The null hypothesis was that the summer of 2003 would not be significantly hotter than the other three summers. The alternative hypothesis was the the air temperatures would be significantly hotter in 2003. A comparison between the results of the Mann-Whitney air temperature and Mann-Whitney air temperature-UHII tests would show if air temperatures were significantly hotter in the heatwave summer of 2003, and if such a difference existed, whether this was captured by the UHII metric.

Furthermore, in order to perform a like-for-like comparison of UHII values derived from AVHRR EST and air temperature data-sets, a modified sub-set of air temperatures was used. The sub-set consisted of UHII values generated using air temperature measurements from the London Weather Centre

(LWC) and High Wycombe stations for the same dates as AVHRR EST scenes. The London Weather Centre Station was chosen as this exhibited the greatest air temperature UHII values for all four summers. The temporal pairing between the MIDAS and AVHRR data meant that air UHII values were only generated for the same dates as those captured by the AVHRR sensor. Furthermore, the use of one urban station (LWC) meant that AVHRR EST and air UHII values for the same location (LWC-HWC) could be compared. Temporal and spatial pairing between AVHRR EST and MIDAS air temperature data was achieved using the database interface to create an SQL query which generated air temperature UHII from LWC and HWC for the dates of the AVHRR scenes and extracted the AVHRR EST UHII values for the pixel location of the LWC station. The paired data were then stored as a new table in the MIDAS PostGIS database, and the minimum, maximum and mean average UHII (LWC-HWC) values from EST and air temperatures of each summer were extracted for comparison.

6.4 Results

6.4.1 Temporal sensitivity of AVHRR estimated surface temperatures

Figure 6.3 shows EST for four AVHRR scenes from 14th August 2002 and 2003 captured within a 65 minute window of each other. For both years, there is a notable increase in EST during the time period of observation (24.46°C at 10:56 am to 26.90°C at 12:56 pm for 2003 and 21.22°C at 10:59 am to 21.77°C at 12:08 pm for 2002), although the magnitude of the rise is greater for 2003 (1.2°C hr^{-1}) when compared with 2002 (0.48°C hr^{-1}). This more dramatic temperature increase for the heatwave year is supported to some extent by the corresponding MIDAS air temperature data, which revealed an average increase in air temperature for London weather stations of 1.68°C for 2003 compared with 0.35°C for 2002 over the same time period. It may be tempting, therefore, to consider that the AVHRR data capture the diur-

nal progression of surface temperature dynamics and differentiate between summers with significantly varying temperatures.

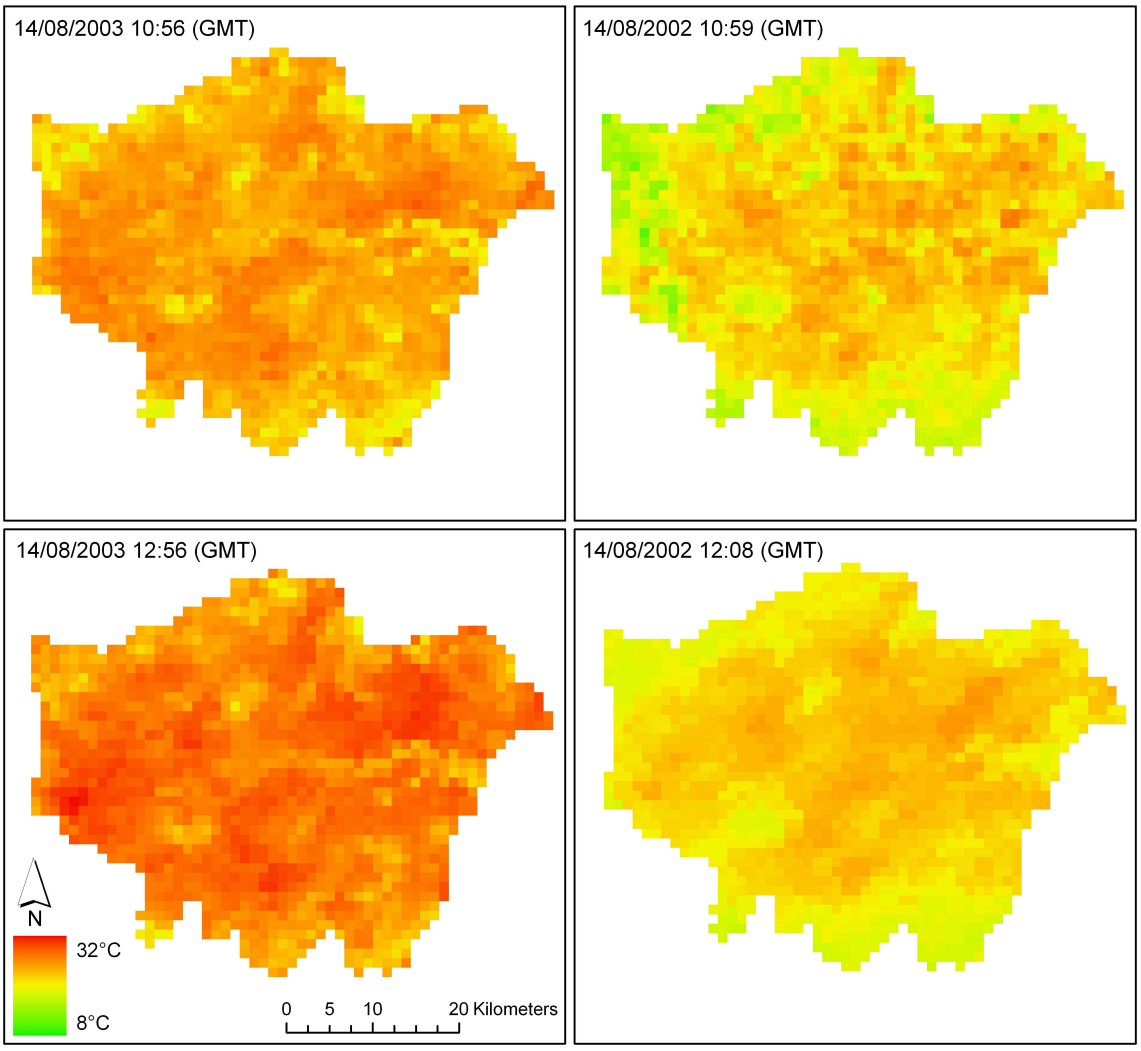


Figure 6.3: AVHRR estimated surface temperatures (ESTs) for London scenes acquired on 14 August 2003 and 2002. *Note: all scenes are contrast stretched to the same minimum (8°C) and maximum (32°C) values for consistency.*

However, Table 6.3 reveals a more complex relationship exists between individual scenes (within 65 minutes of a 2003 scene) and daily averages of ESTs for the 75th-percetile (1997), median (2001), and 25th percentile (2002) years compared with the 2003 heatwave year. Both individual scenes and daily mean differences show little correlation with percentile ordering of the summers in terms of their overall MIDAS average summer air temperatures. For example, mean scene differences between 2003 and the 75th percentile year of 1997 range from 1.04°C to 9.64°C, which is greater than that of the 25th percentile year of 2002 (-0.04°C to 5.11°C). At a daily level, both 1997

(75th percentile year) and 2001 (50th percentile) have noticeably larger absolute mean differences (5.66°C and 5.19°C, respectively) than 2002 (25th percentile year: 2.04°C). Equally, in the case of daily maximum differences, the 25th percentile year is closer to the heatwave year (9.12°C daily maximum difference) relative to the 75th percentile year (1997: 13.79°C daily mean maximum difference) and 50th percentile year (2001: 10.55°C daily mean maximum difference). One would expect the greatest difference for individual scenes and daily averages to be for the 25th percentile year (2002) and the smallest for the 75th percentile year (1997). This suggests that comparison of AVHRR ESTs at a single scene and daily diurnal level is an unreliable means by which to compare summer temperature dynamics of different years. A probable explanation of these results is that they are strongly influenced by meteorological signals (Tran et al., 2006) and, in the case of daily averages, the limited number of scenes available (Nichol, 2003). In combination, these results imply that daily averaging is too fine a temporal quantisation for the analysis of estimated summer surface temperatures on a year-by-year basis.

Table 6.4 shows the AVHRR EST UHII temporally averaged to the monthly level. In general the monthly data seem to exhibit a slightly better than expected pattern of distribution than the individual scenes. Firstly, apart from June, the monthly average AVHRR ESTs follow the expected pattern of the percentile years derived from the analysis of MIDAS air temperatures from London weather stations (Section 6.3.1). Only June 2001 falls outside this ordering with a mean EST of 26.79°C. The stability of the percentile year ordering is not, however, as evident in the mean minimum and maximum monthly temperatures (Table 6.4), showing a greater monthly variability for cooler years, such as 2001 and 2002 which exhibit quite high mean minimum and maximum monthly ESTs (e.g. June 2001 mean minimum of 17.98°C (3.00°C > 2003) and mean maximum of 32.15°C (4.87°C > 2003)).

2003 scene	2003 scene time	Comparison scene	Time difference (minutes)	Mean difference (°C)	Maximum difference (°C)	Daily absolute mean difference (°C)	Daily absolute mean maximum difference (°C)
8 September 2003	12:12	8 September 1997	64	1.04	11.80	5.66	13.79
8 September 2003	13:52	8 September 1997	-36	6.31	15.18		
8 September 2003	13:52	8 September 1997	65	9.64	14.39		
24 June 2003	11:48	24 June 2001	65	-6.06	8.32	5.19	10.55
14 August 2003	12:56	14 August 2001	-34	4.32	12.78	2.38	9.12
14 August 2003	10:56	14 August 2002	4	3.40	15.06		
14 August 2003	12:56	14 August 2002	-48	5.11	10.22		
14 August 2003	12:56	14 August 2002	53	0.97	6.85		
14 August 2003	14:37	14 August 2002	48	-0.04	4.35		

Table 6.3: Individual and daily scene differences between AVHRR estimated surface temperature (EST) for 2003 (heatwave) and 1997, 2001, and 2002 summers.

Month/year	No. of scenes	Mean ($\bar{x}$) EST (°C)	Mean minimum EST (°C)	Mean maximum EST (°C)
June 2003	9	22.17	14.98	27.28
June 1997	3	20.52	13.92	24.61
June 2001	4	26.79	17.98	32.15
June 2002	8	21.78	15.98	26.12
July 2003	11	27.55	21.54	31.94
July 1997	Too few scenes to generate monthly statistics			
July 2001	4	22.90	17.14	27.05
July 2002	4	17.17	10.98	24.18
August 2003	23	26.35	21.52	29.62
August 1997	Too few scenes to generate monthly statistics			
August 2001	4	24.50	18.90	28.25
August 2002	5	21.56	15.12	26.96
Summer 2003	43	25.94	20.75	29.46
Summer 1997	3	25.31	19.32	29.62
Summer 2001	12	25.83	19.84	30.32
Summer 2002	17	21.07	16.16	24.93

Table 6.4: Monthly and summer spatial averages of AVHRR EST for the Greater London Authority for 2003, 1997, 2001 and 2002 summers derived from AVHRR data.

When further averaging is performed to the full summer season level (1st June to 31st August; Figure 6.4), the pattern of mean temperature being consistent with the weather station percentile ordering persists (Table 6.4). Again, the mean minimum and maximum yearly ESTs show a less consistent relationship, and in the case of the mean maximum estimated surface temperatures, the similarity of the values perhaps suggests that it may be problematic in distinguishing extreme temperature differences at the summer level.

However, the Mann-Whitney U-tests revealed that for all 2003 months the null hypothesis could be rejected as they were statistically hotter that their corresponding month in 1997, 2001 and 2002 at a 95% significance level (Table 6.5). This implies that it is possible spatially on the basis of the overall distribution of ESTs at a monthly level to distinguish the heatwave year as the statistical test applied is pair-wise based on corresponding pixel locations. In a similar manner, at the level of averaging over an entire summer year, all

but 2001 (the median year) were found to be statistically cooler than 2003 at a 95% significance level (Table 6.5).

Year	**U_{min} calculated one-tail X > Y**	**Reject null at 95%**
June 2003-1997	713,661	Yes
June 2003-2001	239,243	Yes
June 2003-2002	1,039,891	Yes
July 2003-2001	119,851	Yes
July 2003-2002	1,835	Yes
August 2003-2001	498,324	Yes
August 2003-2002	108,144	Yes
Summer 2003-1997	988,104	Yes
Summer 2003-2001	1,113,169	No
Summer 2003-2002	98,090	Yes

Table 6.5: Mann-Whitney U-test statistical results comparing monthly and yearly AVHRR ESTs. *Note: reject null if U_{min} calculated < U_{min} critical (1,110,722 for one-tail test X > Y at 95%; X= 2003 heatwave summer).*

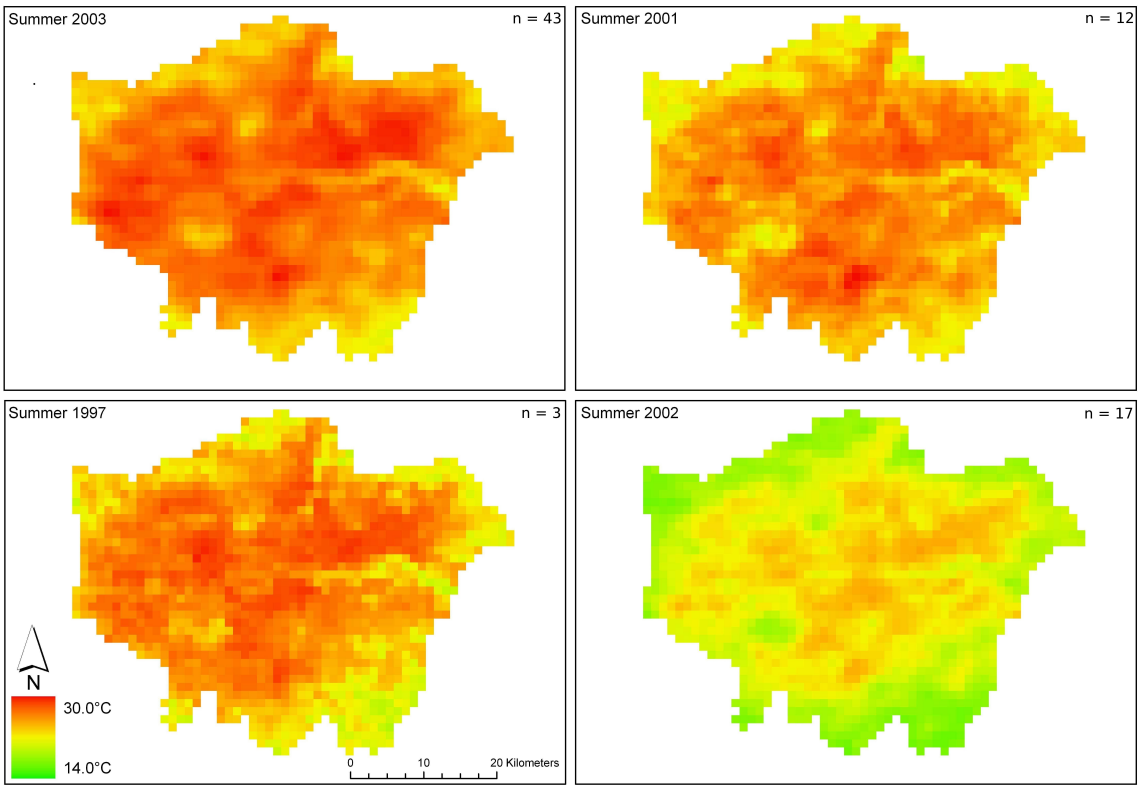


Figure 6.4: AVHRR summer ESTs for the Greater London Authority with temporal averaging at the seasonal level, the number of scenes for each summer is denoted by 'n'. *Note: all scenes are contrast stretched to the same minimum (14°C) and maximum (30°C) values for consistency.*

The importance of choosing the correct temporal and spatial aggregation level is further highlighted in Table 6.6, where the summer season average ESTs of individual pixels containing the urban and rural weather station locations are presented. This shows that at the summer season level there is relatively little agreement between the ordered magnitude of overall mean, mean minimum and mean maximum ESTs compared with the order of screen-level air temperatures (hottest 2003, 75th percentile 1997, 50th percentile 2001, lower percentile 2002). For most weather stations, the 50th percentile (2001) has the highest overall mean and mean minimum temperatures (i.e. SJP, NTH and LWC). More encouraging in this case is the fact that the hottest mean maximum temperature is recorded for the heatwave year of 2003 for all weather station locations, although again for the other years there is an inconsistent ordering. Thus, while Table 6.5 shows that all years apart from 2001 were statistically cooler than 2003 at the summer season level of aggregation, the results of Table 6.6 raise doubts as to whether averaging over an entire summer allows inter-annual temperature dynamics to be captured reliably.

	Station	**SJP**				**LHR**				**NTH**				**LWC**				**HWC**			
Year	No. of scenes	Min	Max	$\overline{x}$ (°C)	σ (°C)	Min	Max	$\overline{x}$ (°C)	σ (°C)	Min	Max	$\overline{x}$ (°C)	σ (°C)	Min	Max	$\overline{x}$ (°C)	σ (°C)	Min	Max	$\overline{x}$ (°C)	σ (°C)
2003	43	10.60	34.91	25.81	4.57	11.31	39.91	28.98	5.30	7.11	37.31	25.52	5.21	11.84	38.64	27.34	5.00	7.38	32.31	21.02	6.32
1997	3	16.31	25.26	21.10	3.68	17.73	29.41	23.32	4.78	16.59	25.82	20.81	3.81	16.89	26.67	22.65	4.18	13.35	19.49	16.19	2.53
2001	12	19.87	31.56	26.43	3.55	18.81	34.72	27.74	5.05	20.89	32.66	26.53	3.90	21.63	32.96	28.23	3.50	10.13	25.08	19.81	4.07
2002	17	14.67	28.37	22.25	3.70	5.66	27.87	21.93	5.42	6.00	25.07	19.42	4.82	12.66	30.88	23.99	4.45	7.55	20.87	15.01	4.15

Table 6.6: Summer seasonal AVHRR EST values from pixel locations containing urban and rural weather stations. *Stations; London Weather Centre: LWC, London Heathrow: LHR, Northolt: NTH, St James's Park: SJP, High Wycombe (rural): HWC.*

6.4.2 The effect of variations in sample size from AVHRR data on estimated surface temperatures

As described in Section 6.2.1 the data selection and pre-processing methodology employed to derive EST from AVHRR data resulted in varying availability of AVHRR scenes for analysis from the four years under consideration (e.g. due to cloud contamination). Therefore, when assessing the temporally aggregated data (e.g. Figure 6.4) the varying sample size used to derive the averages should be taken into consideration (2003: 43 scenes, 1997: 3 scenes, 2001: 12 scenes, 2002: 17 scenes).

Variations in sample size could lead to the mis-representation of a summer's temperature, when scenes are temporally averaged to the seasonal level. For example, if a limited number of scenes for one summer were all captured during warmer than average conditions, temporally aggregated ESTs for that summer could be biased. This is particularly relevant when examining the ESTs from the summer of 1997, for which only three scenes were available. A qualitative comparison between the summer season ESTs in Figure 6.4 appears to show that 1997 exhibits an increased level of spatial heterogeneity in EST as compared to the other three summers. This is potentially the result of the larger sample sizes in the other three years, capturing a greater range of temperatures in each pixel, leading to a more spatially homogeneous representation of average summer ESTs.

Furthermore, the difference in heterogeneity appears to propagate into the derived summer UHII scenes (see Figure 6.7), although, the general spatial pattern of intensity across the city remains the same in all four summers (see Section 6.4.3). Given that sample size variability is an inherent limitation in Earth observation, the differences in availability of scenes between summers further highlights the requirement for a long-temporal baseline of data, to maximise the number of scenes available for analysis.

6.4.3 Analysis of urban heat island intensity metric

While the results from the temporal sensitivity assessment of AVHRR EST data above (Section 6.4.1) demonstrate the utility of monthly ESTs to capture urban temperature dynamics, it is common to employ the UHII metric in order to express the UHI phenomena (Jones and Lister, 2009; Tomlinson et al., 2012). Again, at the individual scene level (Figure 6.5), AVHRR-derived UHII seems to capture the spatial variability in temperature, as previously shown with EST in Figure 6.2. In particular, analysis of Figure 6.5 and the land cover classification shown in Figure 6.6 shows that large urban green areas such as Richmond Park (south west London) and Hampstead Heath (north west London) have some of the lowest UHII values in the city (~ -1.0 to 0.5°C), although intensity in these areas is not as low as regions at the urban-rural fringe (e.g. south of Bromley: ~ -1.5°C). The River Thames also exhibits low intensity values (~ -1.5 to 0.5°C) which can be clearly seen east of Isle of Dogs where the river is at its widest. The locations of these cool islands suggests that the UHII is capturing spatial variation in intra-urban temperatures caused by natural cooling processes (e.g. latent heat loss via evapotranspiration from vegetation) (Oke, 1987; Eliasson and Svensson, 2003; Nichol, 2005).

Furthermore, Figures 6.5 and 6.6 show much greater UHII values in high density built-up areas (~ 6.0 to 8.8°C) which is characteristic of the urban heat island (see Section 2.3 and Table 2.3). The dense urban and industrial regions of the East End above London Docks, and Dagenham show the highest intensities (up to 8.8°C), as do the suburbs of Croydon and Tooting to the south, all of which can be identified in Figure 6.5. In addition to the single scene presented in Figure 6.5 it can be seen that the locations of the identified hot-spots and cool islands are persistent throughout the time-series and can also be seen in seasonal averages of EST (e.g. Figure 6.4), highlighting the importance of characterising UHII in a spatially explicit manner so as to capture the spatial variability seen in the EST values.

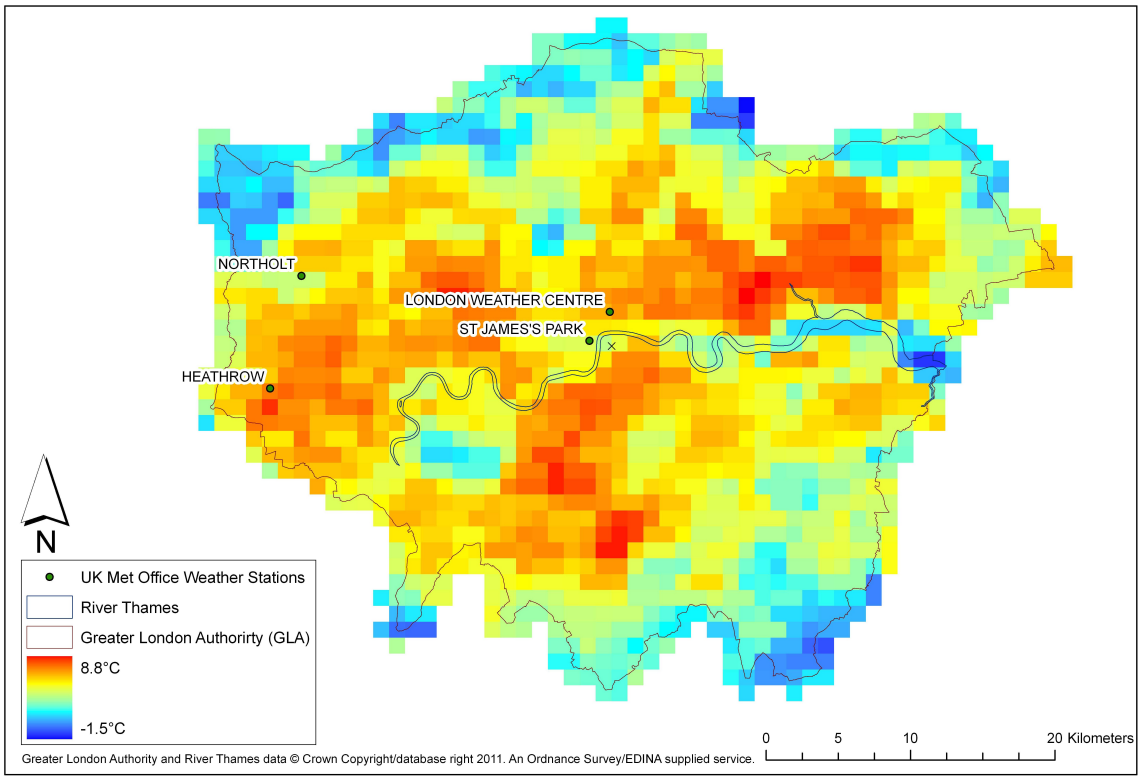


Figure 6.5: Urban Heat Island Intensity (UHII) for London using EST from AVHRR data acquired on 8th August 2003, 14:04 (GMT).

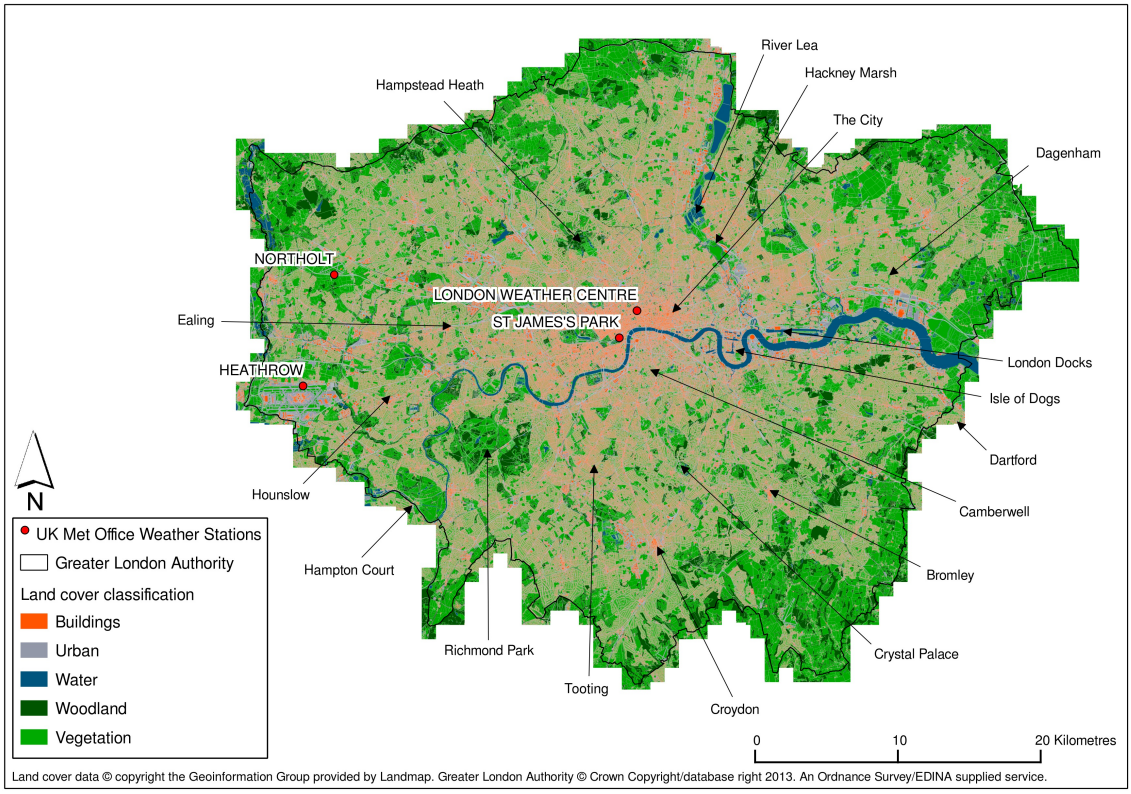


Figure 6.6: Land cover classification and key areas of interest for the Greater London area. *Land cover data from the UKMap and National Land Use Database.*

Year	Mean ($\bar{x}$) air UHII (°C)	Minimum air UHII (°C)	Maximum air UHII (°C)	Mean ($\bar{x}$) EST UHII (°C)	Minimum EST UHII (°C)	Maximum EST UHII (°C)
2003	4.74	3.10	8.30	6.31	1.20	19.77
1997	3.73	3.60	4.00	6.46	3.54	8.66
2001	4.35	2.50	5.90	8.80	1.66	22.83
2002	4.97	2.90	9.00	8.98	3.05	18.21

Table 6.7: Summer season UHII values derived using BADC air temperatures from LWC and AVHRR ESTs from the pixel location containing the LWC weather station.

However, at the location of the weather stations AVHRR EST UHII values were found to be greatest annually in 2002 (the lower quartile summer) at 8.38°C (LWC-HWC) compared with 7.41°C (LHR-HWC) for the heatwave year of 2003. Nonetheless, spatially maximum AVHRR EST UHII values were found to be consistently higher at the LWC in the centre of London for all years except for 2003 where they are found at London Heathrow (LHR) on the urban-rural fringe of London. This suggests that at the intra-annual scale, the AVHRR EST UHII captures a change in the spatial dependence of maximum UHII.

In the case of air temperature UHII, similar results were found, with maximum UHII being captured at LWC over all the four years but with a marked increase in UHII at LHR during the heatwave summer of 2003, resulting in a marginal difference between LWC and LHR of 0.1°C. However, Table 6.7 shows for the location of the LWC that 2003 (the hottest and heatwave year) does not exhibit either the highest maximum or the highest mean UHII for either weather station air temperature (highest maximum and mean occurs for 2002, the 25th percentile year) or AVHRR EST UHII (highest maximum occurs for 2001 (the median year) and highest mean occurs for 2002). Figure 6.7 shows that the AVHRR-derived seasonal average UHII values for each of the four summers and exemplifies this fact. The figure also shows that spatially much higher UHIIs are pervasive in 2002 compared to 2003.

As in the case of AVHRR ESTs, Mann-Whitney U-tests were performed between the 2003 UHII values and the other years investigated for monthly and

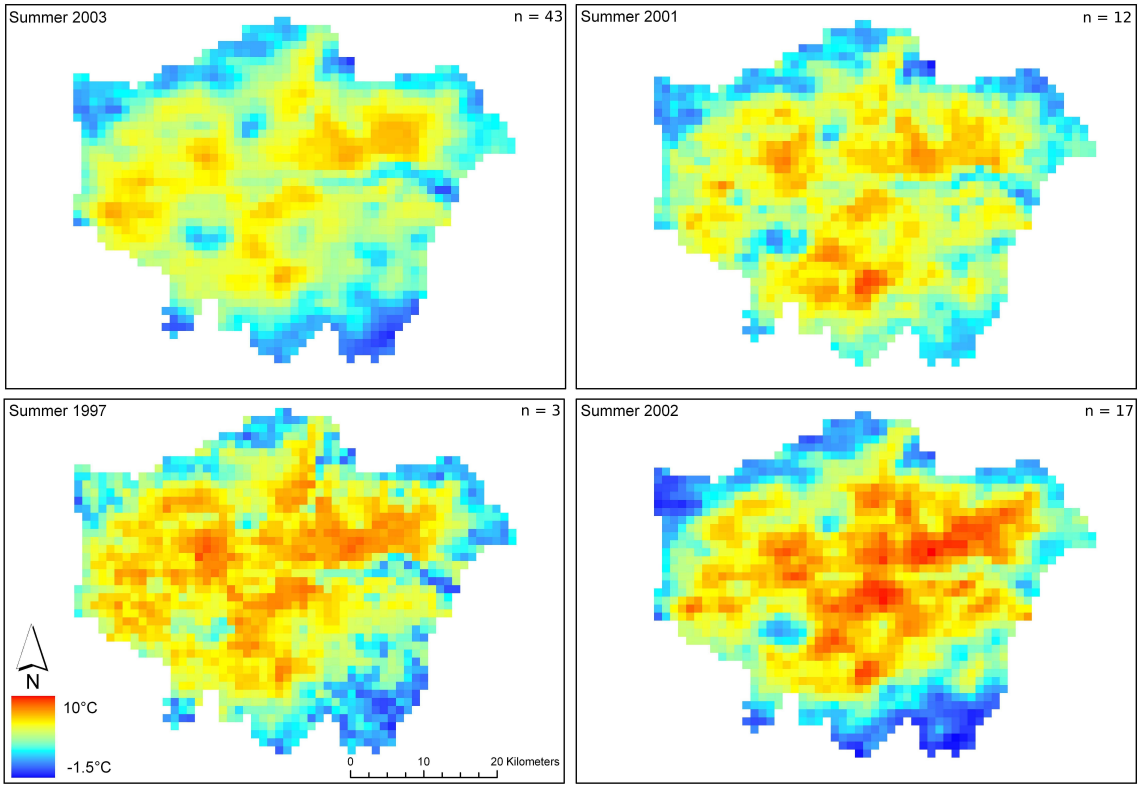


Figure 6.7: AVHRR summer UHII values for the Greater London Authority with temporal averaging at the seasonal level, the number of scenes for each summer is denoted by 'n'. *Note: all scenes are contrast stretched to the same minimum (-1.5°C) and maximum (10°C) UHII values for consistency.*

yearly averages (Table 6.8). These results revealed that at the 95% confidence level only four of the seven pair-wise combinations of AVHRR UHIIs could reject the null hypothesis and accept that 2003 exhibited statistically greater monthly AVHRR EST UHII values than the other summer. It is worth noting that three of the four occur for the month of June.

For the other months (July and August), three out of the four tests found no statistically significant difference between 2003 and the other summers (Table 6.8). It should be noted that during the exceptionally warm summer of 2003, the greatest temperatures were record in August (Johnson et al., 2005); which in association with these results highlights the potential inability of the UHII metric to capture significant temperature changes at the monthly level. At the yearly level of aggregation, the inability to distinguish between the heatwave year and the other years on the basis of the UHII metric is even more striking with all three pair-wise combinations being found to have no statistical difference at a 95% significance level (Table 6.8).

Year	**U_{min}calculated one-tail X > Y**	**Reject null at 95%**
June 2003-1997	313,792	Yes
June 2003-2001	1,034,638	Yes
June 2003-2002	777,175	Yes
July 2003-2001	2,139,493	No
July 2003-2002	991,436	Yes
August 2003-2001	1,169,920	No
August 2003-2002	1,654,360	No
Summer 2003-1997	1,425,958	No
Summer 2003-2001	1,307,869	No
Summer 2003-2002	1,641,385	No

Table 6.8: Mann-Whitney U-test statistical results comparing monthly and yearly UHII values for London derived from AVHRR EST. *Note: Reject null if U_{min} calculated < U_{min} critical (1,110,722 for one-tail test X > Y at 95%; X = 2003 heatwave summer).*

Table 6.9 shows the results of the Mann-Whitney U-test for summer air temperature UHII values. This test showed that the null hypothesis could not be rejected for any of the three pairs of summers at the 95% confidence level. This means that the daily UHII values generated using air temperatures from the heatwave summer of 2003 are not significantly greater than the UHII values from any of the other three cooler summers (Table 6.9).

Years	**U_{min} critical (one-tail)**	**U_{min} calculated one-tail X > Y**	**Reject null @ 95%**
2003-1997	3073	3600	No
2003-2001	3273	3825	No
2003-2002	3394	3960	No

Table 6.9: Mann-Whitney U-test statistical results comparing daily summer UHII values for London derived from MIDAS screen-level air temperatures. *Note: Reject null if U_{min}calculated < U_{min}critical (one-tail test X>Y at 95%; X = 2003 heatwave summer).*

In contrast, however, Table 6.10 shows the results of the Mann-Whitney U-test for daily averaged air temperatures from the four urban stations, in which it can be seen that the 2003 summer values are significantly greater than all of the other three summers. The discrepancy between the UHII results in Table 6.9 and air temperature results in Table 6.10 indicate that the UHII metric is not capturing an increase in intensity between urban and rural temperatures during the significantly warmer 2003 summer. This suggests that either the UHII metric is insensitive to the changes which are taking place, or there are no significant changes in the urban-rural temperature difference during the heatwave summer.

Years	**U_{min} calculated one-tail X > Y**	**Reject null @ 95%**
2003-1997	3488	Yes
2003-2001	3072	Yes
2003-2002	2537	Yes

Table 6.10: Mann-Whitney U-test statistical results comparing daily summer screen-level air temperature values averaged over the four London stations (LWC, LHR, NTH and SJP). *Note: Reject null if U_{min} calculated < U_{min} critical (3637 for one-tail test X > Y at 95%; X = 2003 heatwave summer).*

The above analysis of the UHII values using AVHRR EST values reveals that it is possible to capture at a single point in time the spatial representation of the UHI. However, the results also pose serious questions about the utility of the UHII metric when derived using either thermal Earth observed data or terrestrial screen-level air temperatures. The data show that there is no significant change in the difference between urban and rural temperatures during a heatwave summer, resulting in the metric's inability to quantify the changes between heatwave and non-heatwave summers. In the case of the AVHRR UHII results, such differences may be the result of vegetation loss at the rural station during the heatwave summer due to drought, which will change the cover-surface temperature interaction (Lu and Weng, 2006) and cause the metric to improperly quantify the 'true' UHI.

However, this does not explain the similar observations from corresponding weather station screen-level air temperature-UHII results (Table 6.9). Taken together, Table 6.7 suggests that the UHII derived either from AVHRR scenes or from weather station records is questionable in capturing intra-annual temperature dynamics and, in particular, heatwave events. Interestingly, similar questions have been raised when using solely air temperature-derived UHII (Jones and Lister, 2009).

While the AVHRR data represent daytime scenes, the comparative air temperature-based UHII values in Table 6.7 are generated from hourly values over the full diurnal cycle at each weather station, suggesting that the above differences in UHII between summers are manifested in both daytime surface and diurnal air temperature based UHII observations. This further supports the evidence that UHII is not a suitable metric for the analysis of urban temperature dynamics within the remit of urban climate and extreme temperature event analysis.

6.5 Conclusions

The research presented in this chapter has demonstrated how a long temporal baseline of daytime AVHRR data can be employed to capture the summer temperature regime of London, UK, including the response to a known heatwave event. By processing a large number of daytime scenes for a sample of years that characterise the temperature distribution of London over a decade period, it has been possible to evaluate the intra-annual temperature dynamics of London. The results presented show, as in other studies (e.g. Tomlinson et al. (2012)), that both temporally and spatially significant surface temperature variations can be captured using thermal Earth observation data.

By evaluating temporal averaging at the daily, monthly, and summer season level, it has been shown that there is, in the case of London, a high level of sensitivity in the AVHRR data to diurnal and localised meteorological effects (Nichol, 2003; Tran et al., 2006). Equally at the other end of the level aggregation, it was found that summer season averages fail to capture the ordered magnitude of inter-annual temperature dynamics recorded by established weather station screen-level air temperatures. The analysis reveals that characterising ESTs at the monthly level gives the best statistical discrimination of the inter-annual temperature of London and allows a heatwave summer year to be distinguished. The results presented highlight the importance of generating robust temporal averages from multiple scenes to remove noise and quantify the underlying urban temperature regime.

With regard to the UHII metric, the research has demonstrated the utility of AVHRR scenes to generate UHII surface maps which represent spatial variability of intensity over the urban surface. However, the utility of the metric is questioned due to the lack of response seen in the UHII data derived from both AVHRR EST and air temperature measurements between different summer temperature regimes. Testing of these data in a statistically robust manner showed that the 2003 heatwave UHII data sets, for both EST and screen-level air temperatures, did not exhibit significantly greater intensities than the other three years under consideration.

However, several other studies have reported distinctive UHII values during heatwaves for several cities (Cheval and Dumitrescu, 2009; Cheval et al., 2009; Tomlinson et al., 2012). As such, further research is required before a definitive evaluation of the utility of the UHII metric can be made. Nonetheless, on the basis of the results presented here for both AVHRR ESTs and weather station screen-level air temperatures, along with the corresponding findings of other studies (e.g. Jones and Lister (2009)), a more suitable metric for daytime AVHRR scenes to characterise temporally, urban temperature dynamics and perform intra-annual comparisons may be based on summer seasonal monthly average or monthly average maximum temperatures.

Chapter 7

Discussion and analysis

7.1 Introduction

This chapter presents a discussion and analysis of the findings of the research undertaken in this project in relation to the thesis aim and objectives (see Section 1.3). The extensive literature review presented in Chapter 2 showed that intra-urban spatial variability of exposure to the heat hazard was key in determining the risk of increase in mortality and morbidity to vulnerable urban populations during heatwave events (e.g. Harlan et al. (2006); Dousset et al. (2011)). A number of studies have utilised thermal Earth observation to capture spatially-complete estimates of surface temperature as a supplement or alternative to spatially-sparse terrestrial air temperature measurements to quantify intra-urban temperature variability (e.g. Roth et al. (1989); Streutker (2002); Dousset and Gourmelon (2003); Fung et al. (2009)).

However, the literature review identified two key research challenges which need to be addressed to better support the use of thermal Earth observation to quantify intra-urban spatial variability of heat hazard exposure during extreme temperature events. First, long time-series of thermal Earth observed EST and air temperature measurements are required to investigate the ability of empirical models to generate spatially-complete estimates of air temperature. Second, the utility of thermal Earth observation to quantify the

magnitude of a heatwave summer using EST and the UHII metric needs to be evaluated. As such, the software development and analysis presented in Chapters 4-6 aimed to address these challenges using a long time-series of integrated AVHRR EST and MIDAS terrestrial air temperature measurements. In this chapter, Section 7.2 reports on the feasibility of using long time-series of integrated AVHRR and MIDAS measurements to derive empirical models of intra-urban air temperatures, and Section 7.3 summarises the findings from analysis of the UHII metric during a heatwave event. Section 7.4 reports some final comments.

7.2 The EST-air temperature relationship

7.2.1 Spatial variability in EST and air temperatures

The analysis conducted in Section 5.4.1 investigated the relationship between EST and air temperatures. The results showed differences in the spatial-variability of EST and air temperatures as measured at the four London weather stations. Over the time-series of summer temperatures (1985-2008) four of the six pairwise station combinations showed significant difference (p=0.05) in EST, whilst only one pair showed significant differences (p=0.05) in air temperature. These results suggest that Earth observed EST values are capturing spatial variability not present in the air temperatures as measured at the four weather stations. Previous studies have shown similar results in that ESTs have greater variability than air temperatures (Kawashima et al., 2000; Nichol et al., 2009). For example over a 2.5km transect in central Hong Kong, Nichol et al. (2009) showed EST variations greater than ~±2.5°C, but air temperature variations measured over the same transect were less than ~±1.5°C. Potentially the results presented in Section 5.4.1 and those from the literature indicate that air temperature doesn't necessarily capture intra-urban variability in temperatures to the same degree at EST.

Testing for significant differences between EST and air temperatures at the

global level (using an average from the four weather stations) showed a significant difference between the two (see Section 5.4.1). However, testing at the station level revealed a more complex relationship, with two of the stations (SJP and NTH) exhibiting statistical similarity between EST and air temperatures, and the remaining two stations (LWC and LHR) showing no similarity. The literature shows that urban surface properties control urban EST and air temperatures (Dousset and Gourmelon, 2003; Eliasson and Svensson, 2003; Nichol, 2005), and that the EST-air temperature correlation increases with vegetation density (Kawashima et al., 2000). Given that SJP and NTH represent areas with vegetative cover and LHR and LWC represent urban land cover, one explanation for the differences seen in relationship between EST and air temperatures may be the land cover at each station. Further work examining the relationship between EST and air temperature using known land cover for the station locations would help confirm whether this is the case.

In conjunction with these results, the level of correlation between EST and air temperatures at the global level (using an average from the four stations, r^2= 0.26) was much lower than that of spatially-paired EST and air temperatures from the four stations (r^2= 0.51) (see Section 5.4.1), and suggest that the relationship between EST-air temperatures is correlated with location. To some extent this is to be expected as a number of studies have shown the influence of location on the relationship between intra-urban temperatures (e.g. Czajkowski et al. (2000); Fung et al. (2009); Nichol (2009)) and in particular that the strength of the relationship changes between urban (r^2= 0.42) and rural (r^2= 0.09) land cover types (see Table 2.9, Nichol (2009)). However, the analysis conducted in Section 5.4.1 directly addresses a limitation of previous studies, by using a long time-series of integrated EST and air temperatures as opposed to a limited number of scenes with which to test for spatial variability in the relationship. As such, the implications of the findings in Section 5.4.1 mean that future research investigating the relationship between EST and air temperatures needs to use spatially-paired measurements to properly capture the relationship between the two.

7.2.2 Modelling air temperatures

Section 5.3.1 focused on the use of the empirical relationship derived between spatially paired summer daytime EST and air temperature measurements at the four London stations. Using an expanded data-set to improve temporal coverage in daytime scenes, correlation between the two data-sets (r^2= 0.68) was found to be within the expected bounds seen in the literature (r^2= 0.45-0.82 (Vogt et al., 1997; Nichol, 2009)).

A predictive model of air temperature was derived from the linear regression of daytime (06:00-21:00) summer (May-September) EST and air temperatures. The model was created as a Python module, building on the software framework developed for data analysis. The model function used the model equation (See Section 5.3.2) to return an estimated air temperature from input EST. Evaluating the model using two-fold and leave-one-out cross validation tests provided RMSE values of 5.03 and 5.02 respectively. The small variation between these two models indicates that the observations are evenly weighted (see Section 5.3.3). Cross-validation tests were performed by removing each of the stations in turn and over selected summers showed promising results. For cross-validation of the individual stations, the maximum RMSE ranged from 2.70 to 2.94. The low variation (0.24) between these results indicate that the relationship between EST and air temperatures across the four stations is spatially robust. This means that each triplet combination of stations can be used to derive a model capable of estimating the air temperature for the removed station with a little over half the RMSE of the standard two-fold cross validation. These results show that an empirical model of air temperatures derived from four weather stations is sufficient to form spatially-robust estimates of air temperature and as such could potentially be used to create a model to estimate spatially-complete air temperature measurements across the urban surface.

Additionally, the cross-validation testing for individual summers was equally promising, with RMSE values lower than the two-fold test, ranging from 2.51 to 3.11. The results showed no apparent relationship between RMSE and av-

erage summer temperatures. For example the hottest and coolest summers (1995 and 2002) had the lowest RMSE values (2.51 for both). These results indicate that the model is capable of predicting temperatures over a range of summers in the time-series which are independent of the summer temperature. As such the results validate the utility of using a long time-series of thermal Earth observation data (Roth et al., 1989; Voogt and Oke, 2003) as opposed to a single or limited number of scenes. This enables the model to capture temperatures over a range of summers in order to derive valid estimates during extremes as well as average years.

The spatio-temporally paired EST and air temperature data used in the model was also examined for systematic error as a result of the 30 minutes time difference which may occur between satellite overpass and weather station observation time. In this regard the study found no evidence for systematic error as a result of the 30 minute difference in pairing, meaning that small temperature differences which may be as a result of EST measurements being 30 minutes before or after the time of weather station measurements are unlikely to effect the validity of the regression model. This means that future studies can reliably employ this approach for temporal pairing of EST and air temperature measurements, without concern over the 30 minute difference influencing regression analysis of the two.

Section 5.5 presents the results of a series of predicted air temperature scenes across the GLA derived using the model. Using an average of all summer EST values from the time-series of AVHRR data, a spatially continuous estimate of London's air temperature was generated. The modelled air temperatures showed promising results, exhibiting the same intra-urban spatial variability in temperatures as seen in EST (Figure 5.13). For example, surface features which exhibit cooler ESTs such as the River Thames can also be seen to exhibit cooler modelled air temperatures.

To evaluate estimates of air temperature the model was re-created with paired EST and air temperature measurements from each of the four weather stations removed in turn. Figure 5.14 shows the outputs from the four mod-

els, each with data from one of the stations removed. The outputs from this evaluation corresponded with those from the cross-validation results (Table 5.9), and with one station removed the derived model was able to generate spatially-complete air temperatures over the GLA with minimal variation ($\leq$0.12°C difference in standard deviation) compared to the global model (using all four stations). The maximum difference between average predicted air temperature from the global model and the four models with a station removed was 0.61°C.

A second series of predicted air temperatures was created using the global model (based on the regression between EST and air temperatures from all stations and all summers in the time-series) and EST values from six different summers (ESTs were temporally averaged to the summer level). The prediction of air temperatures for this select group of summers was evaluated against averaged air temperatures from the four weather stations over the same summer period. The estimates from the six summers show an average error of 0.69°C (range: 0.23-1.44°C) over the six summers.

The results from the model positively demonstrate the utility of models generated in this way to provide spatially-continuous air temperature estimates to examine intra-urban temperature variations during extreme temperature events, as inputs to future heatwave mitigation and adaptation studies. However, further work is required to evaluate the accuracy of such a model outside of the time-series used to derive the empirical relationship, as future heatwaves are predicted to be more severe and last for longer (Rosenzweig et al., 2011; Hansen et al., 2012) it is likely that such future extreme events will be less well characterised by the empirical model presented here. Additionally, evaluation of the model could be extended to test the accuracy of predictions using air temperature measurements from additional weather stations.

7.3 Earth observed urban temperature dynamics

7.3.1 Temporal sensitivity of thermal Earth observation

To perform analysis of AVHRR temporal sensitivity and examine London's urban heat island using AVHRR ESTs, a sub-set of the available time-series was created to scale the number of scenes used, and reduce the processing overhead. The subset was created in a manner so as to reduce the number of scenes for processing but to ensure that the general temperature record of London was still captured, with a suitable number of scenes for analysis (Voogt and Oke, 2003; Cheval and Dumitrescu, 2009; Nichol et al., 2009). The sub-selection consisted of four years representing the median, lower quartile, upper quartile and hottest summers between 1996 and 2006. This method ensured that the selected sample of summers covered the range in distribution of temperatures during this period and allowed testing between summers which were known to exhibit different temperature regimes.

Individual scenes captured on a specific day and within 65 minutes of a 2003 scene, and daily averages of ESTs, from three cooler summers (upper quartile, median and lower quartile) were compared to the expected ordering of percentile summers. The results of this comparison showed no correlation with the expected percentile ordering of the summers. For example whilst the greatest differences were expected between scenes from the heatwave summer (2003) and the lower quartile summer (2002), this was not the case, and the maximum difference was found with the upper quartile summer (1997). Therefore, whilst the utility of individual scenes for spatio-temporal pairing with air temperatures was demonstrated in the previous section, these results suggest that at the individual and daily levels ESTs are not suitable for inter-year comparisons. As discussed in Section 6.4.1, it is likely that this is the result of local meteorological conditions at the time of satellite overpass, which have been shown to influence ESTs (Tran et al., 2006; Tomlinson et al., 2012). Such variability could be investigated further by grouping scenes by

weather conditions at the time of overpass, and examining scenes within the same classification, in a similar manner to Tomlinson et al. (2012). These results show that to perform comparisons between intra-urban temperature variations in heatwave and non-heatwave years, individual thermal Earth observed scenes cannot be used.

Temporal averaging showed that ESTs averaged to the monthly level for the four years under consideration exhibited a considerably better pattern of temperature distribution than the individual scenes and daily averages. Except for the month of June, the monthly mean ESTs for all four years exhibit the expected pattern of percentile ordering (e.g. 2003 summer months were hotter than the other summer months). Testing monthly temporal averages from the 2003 (heatwave) summer against the months in the other three summers showed that for all years monthly averages from 2003 were significantly warmer (p=0.05) than the other summers.

The summer season temporal EST averages showed similar results to the monthly averaged ESTs. However, seasonal averages were not as robust as monthly averages in comparison to the expected ordering of the four summers or the Mann-Whitney tests. The season average for the summer of 2001 (the median summer) was warmer (25.83°C) than the upper quartile summer (1997; 25.31°C) and only marginally cooler than the summer of 2003 (25.94°C). The summer of 2003 was shown not to be significantly hotter than 2001 (p=0.05) when temporally averaged to the summer seasonal level.

Furthermore, examination of the seasonal EST values at the locations of the four weather stations revealed that whilst for all stations the mean maximum temperature was greatest in 2003, for the other three years there was inconsistent ordering and that the highest mean temperatures at the weather stations are split between 2003 (LHR, HWC) and 2001 (SJP, LWC) (Table 6.6). These results are of importance to future studies wishing to use time-series thermal Earth observation for inter-year comparisons, as they raise doubts as to whether seasonal averages should be used, and indicate that studies wishing to seek robustness by temporal averaging for inter-year comparisons

should examine ESTs at the monthly level.

7.3.2 Analysis of the urban heat island intensity metric

A number of methods to generate UHII using thermal Earth observation are presented in the literature (see Section 2.4.5). The UHII method proposed by Cheval and Dumitrescu (2009), using a buffer of rural pixels outside the city extents to generate a spatially robust average rural temperature, which is then subtracted from urban pixels, was considered. However, the findings from Cheval and Dumitrescu (2009) and Cheval et al. (2009) indicate that this can introduce ambiguity into the rural temperature record, potentially as a result of non-urban pixels being included in the buffer zones. The method proposed by Streutker (2002), is aimed at inter-city comparisons and uses a Gaussian fit to smooth the data, reducing spatial variability, and as such was not considered suitable for this study. Therefore the urban heat island intensity metric was generated from AVHRR EST data in a method similar to Tomlinson et al. (2012), using the difference between urban ESTs and the EST for the location of a known rural weather station (HWC, see Figure 6.2). This method was selected as it provides the closest interpretation of the UHII metric as applied to full-diurnal rural and urban air temperature measurements. However, based on the results of the temporal sensitivity analysis discussed above, UHII was calculated using monthly and seasonal EST averages, to give the best comparison in intensity between heatwave and non-heatwave summers.

Using a custom function, developed as part of the data processing framework, UHII values derived from AVHRR EST values were generated for the GLA (Figure 6.5). The derived AVHRR EST UHII values show spatial variability in intensity across the urban surface, with cooler features (e.g. the River Thames) exhibiting a lower intensity than warmer areas. These results demonstrate the applicability of long-time series ESTs to generate spatially-complete UHII values to capture intensity across the entire urban surface. There is scope for further research to examine the spatial variability in in-

tensity, and in particular the relationship to urban surface properties. Such information could provide useful information on locations of greatest intensity within the city for heatwave mitigation and adaptation strategies (Harlan et al., 2006; Dousset et al., 2011).

However, despite ESTs at the monthly and seasonal levels showing the 2003 heatwave summer to be significantly hotter than the other three cooler summers, AVHRR EST UHII values derived from monthly and seasonal averages showed ambiguity when tested. Three of the seven monthly averaged EST UHII tests showed no significant difference (p=0.05) in intensity between the 2003 and the other summer in consideration. Additionally, none of the AVHRR yearly averaged EST UHIIs demonstrated a significant difference between 2003 and the other summers. The disparity between statistical tests of EST and UHII values was further supported by results of testing air temperature derived UHII values from 2003 against the other three summers, none of which showed significant difference to 2003. The results showed that the UHII metric generated from either air temperature in the traditional manner or AVHRR EST did not result in a significant increase in intensity during the heatwave summer of 2003. Figure 6.7 demonstrates this, and shows that EST intensity values bear no relationship to summer temperature regime. This suggests that the urban-rural difference isn't significantly increasing during the heatwave summer. These results are key to future studies wishing to examine urban temperature dynamics during heatwave events, as they suggest the UHII metric isn't suitable for such analysis. As such, future research should focus on the development of new metrics, to utilise long time-series thermal Earth observation to examine spatially the change in temperatures during a heatwave event, to identify areas where temperatures are greatest for future heatwave adaptation and mitigation strategies.

7.4 Summary

The requirement for long time-series thermal Earth observed data with which to generate robust EST values for analysis of urban temperatures is a recognised research gap within the literature (Voogt and Oke, 2003). As such, this study has demonstrated the utility of a long time-series of AVHRR ESTs, within a developed software environment, for analysis of London's intra-urban temperature dynamics. Air temperature is a key climatological variable for quantifying urban temperature dynamics, including the relationship between air temperatures and heatwave induced morbidity and mortality (Fish et al., 1985; Johnson et al., 2005; Harlan et al., 2006). However its ability to capture, spatially, temperature dynamics across the urban surface is often restricted by a limited number of weather stations (Cheval and Dumitrescu, 2009; Miller and Small, 2003), information which is required to assess spatial variability in exposure of the urban population to the urban heat hazard (Harlan et al., 2006; Dousset et al., 2011).

This study has demonstrated the utility of a 23-year time-series of integrated EST and air temperature measurements to capture intra-urban temperature dynamics for London during heatwave events. Using the time-series of data it was possible to derive spatially-complete estimates of air temperatures, which are capable of capturing temperatures during heatwave events as well as during average conditions. The output from this model could be used to investigate spatial variability in exposure to the heat hazard during previous heatwave events in a manner similar to Harlan et al. (2006) and Dousset et al. (2011).

It should also be noted that the software framework developed within the project to support data analysis and integration played a critical role in the modelling of air temperature. The software framework enabled the rapid development of modules for model analysis and through its uniform data model enabled predicted air temperature measurements to be output as a series of geospatial raster images for visualisation and interpretation within a GIS environment. The demonstration of the utility of the open source Post-

GIS/GDAL/Python stack for the rapid development of spatio-temporal analysis methodologies for temperature measurements in raster and vector form is considered a key finding of this study.

In addition to the aforementioned requirement for spatially-complete air temperature of urban areas, it was necessary to evaluate the utility of the thermal Earth observed AVHRR data to capture, characterise and quantify the magnitude of intra-urban temperatures during a heatwave summer as compared to non-heatwave summers. If such analysis provided positive findings, then in conjunction with the above-described air temperature modelling, this would strongly support the use of thermal Earth observed ESTs to quantify, spatially, increases in temperature intensity, linked to increased exposure to heat, which can occur during heatwaves (Dousset et al., 2011). In this regard, the study evaluated the use of AVHRR ESTs to derive the commonly used urban heat island intensity metric, and test the metric's ability to capture an increase in intensity during a known heatwave summer.

This evaluation showed that AVHRR ESTs are highly sensitive to localised temporal variations, and that for inter-year comparisons, monthly averages of ESTs showed the best correspondence with the known temperature regime of the summers in question. This is a key finding of the study, as it demonstrates that future analysis using thermal Earth observed AVHRR ESTs should evaluate the need for temporal averaging if comparison between temperatures from different years is required. Furthermore, the study showed the successful application of AVHRR ESTs and the PyRaster module in the processing framework to derive the UHII metric in a spatially-complete manner across the city. The derived UHII scenes show spatial variation in intensity, and provide a valuable insight into the intensity of London's heat island across the city. However, examination of the metric derived from both AVHRR ESTs and terrestrial air temperatures showed no significant increase in intensity during the heatwave summer of 2003, despite exhibiting significantly hotter ESTs and air temperatures during this period. As such, this study can state with confidence that the urban-rural difference as quantified by the UHII metric does not show significant change during a heatwave event.

Chapter 8

Conclusions

8.1 Introduction

The aim of this study was to explore the utility of long time-series thermal Earth observation to quantify intra-urban temperature dynamics during heatwave events. To achieve this aim the following objectives were addressed:

1. Identify and review existing methodologies for quantifying intra-urban temperature dynamics using both terrestrial and Earth observed measurements and critically assess their ability to capture changes in urban temperatures during heatwave events.

2. Develop a software framework to enable the spatio-temporal integration, processing and analysis of terrestrial air temperature measurements and thermal Earth observation data.

3. Examine the relationship between near-surface air temperature and Earth observed surface temperatures to evaluate the potential of deriving empirical models to estimate spatially-complete air temperatures during heatwave events using thermal Earth observation.

4. Assess the ability of thermal Earth observation to capture, monitor and quantify changes in intra-urban temperatures and urban heat island intensity during heatwave events.

This chapter explains how each of the objectives was met and presents the main conclusions that can be drawn from the work. Key findings are presented in relation to their potential influence on future research. The final section of this chapter includes ideas for future work and provides a short concluding statement to this thesis.

8.2 Literature review

The first objective was achieved by performing a detailed review of the literature, which found that spatial variability in urban temperatures is key to understanding exposure to heat hazard during heatwave events (Harlan et al., 2006; Dousset et al., 2011). The literature also showed that while existing weather station networks are good at quantifying urban heat island intensity over long time periods, particularly in relation to global climate trends (Jones and Lister, 2009), they were not suitable to capture intra-urban variations required for future heatwave mitigation and adaptation options (Harlan et al., 2006; Dousset et al., 2011). The review revealed that long time-series thermal Earth observation has the potential to characterise intra-urban temperature dynamics during heatwave events and be used to estimate spatially-complete air temperatures (Vogt et al., 1997; Fung et al., 2009). The review concluded that two key research challenges needed to be addressed. First, further research was required to investigate long time-series integration of thermal Earth observed surface temperatures and terrestrial air temperatures, with emphasis on the potential to model spatially-complete air temperatures during heatwave events, providing important data for future urban climatological studies. Second, utility of thermal Earth observation to quantify summer heatwave events needed to be evaluated.

8.3 Software framework

The software framework presented in Chapter 4 was developed to meet objective two. The framework utilised the Python language to develop a uniform data model based on NumPy arrays and interfaces for both thermal Earth observation data and terrestrial observations stored in the MIDAS database. Utilising PostGIS spatial databases for storage of MIDAS data and AVHRR metadata meant that the framework could be used to select and integrate EST and air temperatures spatially and temporally within the uniform data model. This approach enabled the analysis of paired measurements of EST and air temperatures, and the creation of spatially complete modelled air temperatures as presented in Chapter 5. The PyRaster interface was used to develop processing scripts for temporal averaging of AVHRR EST scenes and generate UHII from EST values for analysis during a heatwave event (Chapter 6). In conclusion, Chapter 4 demonstrated the utility of Python and PostGIS to develop tools for data integration and analysis.

8.4 Modelling air temperature

Chapter 5 examined the relationship between AVHRR EST and MIDAS air temperatures, and derived an empirical model to estimate air temperatures across London using EST. Statistical testing of EST and air temperatures showed statistically significant variations between ESTs captured at the location of the four London weather stations. In contrast, air temperature from the same stations was not found to be significantly different. Furthermore, the relationship between EST and air temperatures was statistically significant at stations SJP and NTH, but not LWC and LHR, suggesting that location is a controlling factor in the relationship. As such, correlation between EST and air temperatures was found to be location dependant, with spatially averaged EST and air temperatures giving lower levels of correlation (r^2= 0.26) than non-averaged, spatially-paired measurements (r^2= 0.51).

Using a temporally extended data-set to reduce the influence of outlying, cooler temperatures (r^2= 0.68), a linear regression between EST and air temperatures from the four stations was used to derive an empirical model to estimate daytime air temperature from EST. Two-fold cross validation showed a high model error (RMSE 5.03), however yearly cross-validation testing showed lower more promising error values (RMSE 2.51-3.11). The use of long time-series data was key in this regard as this meant that the data used to form the model covered a range of summer temperatures including the 2003 heatwave summer. As a result, the model was capable of estimating spatially-complete air temperatures during extreme temperature events, such as the 2003 heatwave summer with no discernible increase in prediction error (RMSE 2.94, average absolute error 0.51°C). Furthermore, cross-validation testing also showed that the model exhibited spatial robustness between the locations of the four stations, with any three of the four stations able to define a model capable of estimating air temperatures at the remaining station with an average RMSE of 2.84.

In conclusion, the analysis presented in Chapter 5 demonstrated the utility of long time-series, integrated EST and air temperature modelling of daytime air temperatures in the urban environment. These results are encouraging, and suggest that such data could be used to produce spatially complete estimates of air temperatures which are valid during extreme temperature events and so could potentially be used to aid heatwave mitigation and adaptation research.

8.5 Characterising London's urban heat island during a heatwave event using thermal Earth observation

The analysis presented in Chapter 6 investigated the temporal sensitivity of AVHRR EST to characterise the extreme heatwave summer of 2003 as com-

pared to other cooler summers. Additionally, an evaluation of the UHII metric as generated from EST and air temperatures to capture changes during a heatwave summer was also performed. A sample of four summers from a 10 year period between 1996 and 2006 was used to reduce the number of scenes required for processing and analysis, and ensured that the sample of years covered a range of temperatures including the known heatwave summer of 2003, which was shown to be significantly warmer than the other three summers. Analysis of AVHRR EST sensitivity for inter-year comparison showed that individual scenes captured within 65 minutes of each other and daily average values did not correspond to the exhibited ordering of scenes as based on the known temperatures of each of the sample summers under consideration. However, temporal averages of EST values at the monthly level provided better representation of summer temperatures, with all monthly averages from the heatwave summer of 2003 being significantly warmer than the other three summers. Summer season averages were also generated, but showed a less consistent ordering as compared to known summer temperatures, with average summer ESTs from 2001 (the median summer) exhibiting statistical equality with those from 2003 (the heatwave summer).

Based on the analysis of temporal sensitivity, temporally averaged ESTs at the summer monthly and summer seasonal levels were employed to generated the UHII metric using the difference between urban and rural pixel EST values, in a manner similar to Tomlinson et al. (2012). The generated EST UHII maps showed spatial variability in urban heat island intensity across London, with lower intensities (-1.5-0.5°C) corresponding to low density, vegetated areas and high intensity values (6.0-8.7°C) corresponding to higher density urbanised areas. However, statistical analysis of EST UHIIs showed that the metric only captured intensification between 2003 and the other summers in only four of the seven monthly average combinations. UHII values for 2003 from summer season averages of EST were not found to be significantly greater than summer season average EST UHIIs from any of the other three summers. Additionally, UHII generated with air temperature in the traditional manner also showed no significant increase in 2003 as compared to the

other three cooler summers.

In conclusion the analysis presented in Chapter 6 showed that AVHRR EST scenes are highly variable and that temporal averaging is required to generate values for robust comparisons between summers. Furthermore, UHII generated from temporally averaged ESTs demonstrated the ability to capture intra-urban variations in intensity, not previously quantifiable using terrestrial weather stations. However, the utility of the UHII metric as generated from both EST and air temperatures to capture and quantify significant changes in intensity during heatwave summers is doubtful, suggesting that urban-rural temperature differences are not a consistent means by which to capture and quantify intra-urban temperature changes during a heatwave event.

8.6 Conclusions and future research

The results presented in Chapters 5 and 6 have demonstrated the utility of long time-series thermal Earth observation to investigate intra-urban temperature dynamics, including changes during heatwave events. Specifically, the study has shown that using a custom developed software framework based on the Python language and PostGIS spatial database software it is possible to perform integrated analysis of terrestrial air temperature and thermal Earth observed ESTs in a spatial and temporal manner, and it is suggested that this is an approach which could be feasibly applied in future studies.

The integrated EST and air temperature data were employed to derive an empirical model of spatially complete daytime air temperature based on EST. The model showed promising results when cross-validation testing was used to examine spatial and temporal validity of the model, particularly given its ability to estimate air temperatures during the heatwave summer of 2003 without a noticeable increase in error. These results demonstrate the value of using a long time-series data-set to capture a range of summer temperatures. However, the model could be improved further using higher spatial-resolution

thermal data which have been shown to exhibit a stronger correlation with air temperature (Nichol et al., 2009). Furthermore, additional validation of the model using suitable air temperature data from other weather stations not included in the model could be undertaken to give a better assessment of model accuracy.

Analysis of the ESTs from a selection of summers demonstrated the requirement for temporal averaging of data to provide robust values for inter-year comparisons. Using temporal averaging it was possible to show significant differences in temperature between heatwave summers and cooler years. However, to further understand the drivers of intra-urban temperature variability in London a more thorough investigation of the relationship between land cover and EST needs to be undertaken. Statistical analysis of the relationship between different land cover types (e.g. Figure 6.6) and ESTs through the time series could provide key information for heatwave adaptation strategies and future urban planning.

For example, Hampstead Heath and Richmond Park are large urban green spaces (see Figure 6.6) which form thermal cool islands in the city throughout the time series (Figures 6.4 and 6.7). In contrast, Hackney Marsh is also an area of urban green space of comparable size to Hampstead Heath, yet does not appear to form a definitive cool island. A key research question in this regard is what size of urban green space is required to create significantly lower temperatures than the surrounding city, as measured by the AVHRR sensor.

Furthermore, given the changes in natural land cover which have occurred during previous heatwave events (e.g. vegetation dieback due to drought), and the subsequent changes in thermal properties of the land (e.g. decreased albedo, loss of natural cooling), the time series could be used to investigate the effectiveness of cooling by urban green spaces during known heatwave events. Earth observation potentially offers a major advantage over traditional terrestrial measurements in this regard in that land cover can be captured using data from the visible wavelengths at the same time as EST mea-

surement. Such information would help validate urban greening programmes which have been implemented in a number of cities to promote natural cooling (e.g. Nichol (1994); Solecki et al. (2005)).

Additionally, further analysis should include an assessment of potential cooling by the River Thames which also exhibits low ESTs, in the east of the city, throughout the time series. It would be interesting to test whether pixels adjacent to the river are significantly cooler than elsewhere in the city, as a result of heat absorption and cooling by the river. If a significant difference were found, the results could aid heatwave adaptation and urban planning policy in the increased use of open water in the urban environment to aid cooling.

To better understand spatial variability in exposure to urban temperature extremes captured by AVHRR ESTs the research presented in this thesis could be extended in-line with that of Solecki et al. (2005), Harlan et al. (2006) and Dousset et al. (2011) to link heatwave exposure with vulnerability in order to generate a map of spatial heat risk during heatwave events. To create such a map a GIS environment could be used to overlay the locations of vulnerable populations (e.g. the elderly) on top of ESTs captured during previous heatwaves, to identify the locations at greatest risk in the city (e.g. integration of locations with highest vulnerability and highest exposure to the heat hazard).

Furthermore, extending the risk analysis using records of mortality and morbidity as a result of previous heatwave events (assuming such data were available, data protection issues notwithstanding) would afford testing for correlation between the location of morbidity/mortality and heatwave ESTs. Linking these results with the empirical model of air temperature (see Chapter 5) could yield a relationship between temperature, location and morbidity/mortality. In a process similar to that used to generate heat health watch system metrics (see Table 2.2), geostatistical analysis techniques could then be used to identify patterns in the spatial distribution of heat risk (e.g. spatial clustering or mortality). If EST were available in real-time, such a system could be developed in an operational manner to give a near-real-time evaluation of

current spatial heat risk for the city.

Whilst the derived UHII values presented in Chapter 6 captured spatial variability in intensity, the inter-year comparison of UHII values raised doubts as to the metric's ability to record significant changes in temperature during heatwave events. To integrate the future areas of research discussed above (understanding of thermal drivers and risk analysis), and further our understanding of the urban thermal environment, the development of new metrics which utilise thermal Earth observation to capture spatio-temporal changes in intra-urban temperatures during heatwave events is required.

Previous metrics have focused on quantifying the urban heat island using rural temperature measurements, however, the findings presented in Chapter 6 (see Section 6.4.3) show that new metrics need to focus on developing alternative baselines against which the severity of temperature increases in a city during a heatwave event can be assessed. For this purpose the long time series of integrated EST and air temperature measurements developed in this thesis could be used. Specifically, the intensity of temperature at each location (pixel or weather station) during a heatwave event could be measured by its difference from that of a temperature baseline for the same location generated from the time series. Such a metric would provide a spatially independent measure of heatwave intensity, and for example, during a heatwave event could show how many standard deviations hotter the temperatures in a specific neighbourhood were, as compared to baseline temperatures for the same neighbourhood over the preceding 20-30 year period.

References

ACPA (2005). American concrete pavement association, albedo: A measure of pavement surface reflectance. *Concrete Pavement and Research Technology 3.05*.

Aida, M. (1982). Urban albedo as a function of the urban structure - a model experiment. *Boundary-Layer Meteorology 23*(4), 405–413.

Ascher, D., P. F. Dubois, K. Hinsen, J. Hugunin, and T. Oliphant (1999). *Numerical Python*. Lawrence Livermore National Laboratory.

Baker, L., A. Brazel, N. Selover, C. Martin, N. McIntyre, F. Steiner, A. Nelson, and L. Musacchio (2002). Urbanization and warming of Phoenix (Arizona, USA): Impacts, feedbacks and mitigation. *Urban Ecosystems 6*(3), 183–203.

Balling, R. and S. Brazel (1987). Time and Space Characteristics of the Phoenix Urban Heat Island. *Journal of the Arizona-Nevada Academy of Science 21*(2), 75–81.

Barata, M., E. Ligeti, G. De Simone, T. Dickinson, D. Jack, J. Penney, R. M., and R. Zimmerman (2011). *Climate Change and human health in cities. Climate Change and Cities: First Assessment Report of the Urban Climate Change Research Network*, Chapter 7, pp. 179–213. Cambridge University Press.

Barducci, A. and I. Pippi (1996). Temperature and emissivity retrieval from remotely sensed images using the "grey body emissivity" method. *Geoscience and Remote Sensing, IEEE Transactions on 34*(3), 681–695.

Becker, F. and Z. Li (1990a). Temperature-independent spectral indices in thermal infrared bands. *Remote Sensing of Environment 32*(1), 17–33.

Becker, F. and Z. Li (1990b). Towards a Local Split Window Method Over Land Surfaces. *International Journal of Remote Sensing 11*(3), 369–393.

Betts, R. and M. Best (2004). Relative impact of radiative forcing, landscape effects and local heat sources on simulated climate change in urban areas. *BETWIXT Technical Breifing Note No. 6, Met Office, Exeter, UK*, 15.

Bhattacharya, S. (2003). "European heatwave caused 35,000 deaths." Last modified October 10, 2003. `http://www.newscientist.com/article/dn4259-european-heatwave-caused-35000-deaths.html`.

Blake, R., T. Grimm, R. Horton, S. Gaffin, S. Jiong, D. Bader, and L. Cecil (2011). *Urban climate: Processes, trends, and Projections. Climate Change and Cities: First Assessment Report of the Urban Climate Change Research Network*, Chapter 3, pp. 43–81. Cambridge University Press.

Booth, D. B. and J. Leavitt (1999). Field evaluation of permeable pavement systems for improved stormwater management. *Journal of the American Planning Association 65*(3), 314–325.

Brugioni, D. A. (1983). The Census: It Can Be Done More Accurately with Space-Age Technology. *Photogrammetric Engineering and Remote Sensing 49*, 1337–1339.

Budikova, D., C. M. Hogan, M. Hall-Beyer, G. H. Galal Hussein, and M. Pidwirny (2010). Albedo. *Encyclopedia of Earth. Eds. Cutler J. Cleveland (Washington, D.C.: Environmental Information Coalition, National Council for Science and the Environment). [First published in the Encyclopedia of Earth May 8, 2010; Last revised Date May 7, 2012; Retrieved August 30, 2012. `http://www.eoearth.org/article/Albedo?topic=54300`].*

Burt, S. (2004a). The August 2003 heatwave in the United Kingdom: Part 1 - Maximum temperatures and historical precedents. *Weather 59*(8), 199–208.

Burt, S. (2004b). The August 2003 heatwave in the United Kingdom: Part 3 - minimum temperatures. *Weather 59*(10), 272–273.

Cai, G., M. Du, and Y. Xue (2011). Monitoring of urban heat island effect in Beijing combining ASTER and TM data. *International Journal of Remote Sensing 32*(5), 1213–1232.

Campbell, J. B. (1996). *Introduction to Remote Sensing* (Second ed.). Taylor & Francis.

Caselles, V., C. Coll, and E. Valor (1997). Land surface emissivity and temperature determination in the whole HAPEX-Sahel area from AVHRR data. *International Journal of Remote Sensing 18*(5), 1009–1027.

Cheval, S. and A. Dumitrescu (2009). The July urban heat island of Bucharest as derived from MODIS images. *Theoretical and Applied Climatology 96*, 145–153.

Cheval, S., A. Dumitrescu, and A. Bell (2009). The urban heat island of Bucharest during the extreme high temperatures of July 2007. *Theoretical and Applied Climatology 97*, 391–401.

Clarke, J. (1972). Some effects of the urban structure on heat mortality. *Environmental research 5*(1), 93–104.

Cooper, D. and G. Asrar (1989). Evaluating atmospheric correction models for retrieving surface temperatures from the AVHRR over a tallgrass prairie. *Remote Sensing of Environment 27*(1), 93–102.

Cracknell, A. P. (1997). *The Advanced Very High Resolution Radiometer*. Taylor & Francis.

CRED (2012). Centre for Research on the Epidemiology of Disasters, "EM-DAT The International Disaster Database." Accessed May 5, 2012. `http://www.emdat.be/`.

Czajkowski, K., S. Goward, S. Stadler, and A. Walz (2000). Thermal remote sensing of near surface environmental variables: application over the oklahoma mesonet. *The Professional Geographer 52*(2), 345–357.

Dash, P. (2005). *"Land surface temperature and emissivity retrieval from satellite measurements"*. PhD diss. Universität Karlsruhe.

Dash, P., F.-M. Gottsche, F.-S. Olesen, and H. Fischer (2002). Land surface temperature and emissivity estimation from passive sensor data: theory and practice-current trends. *International Journal of Remote Sensing 23*(13), 2563–2594.

Department of Health (2011). Heatwave Plan 2011: Protecting health and reducing harm from extreme heat and heatwaves. Accessed on May 6, 2012. http://www.dh.gov.uk/en/Publicationsandstatistics/Publications/PublicationsPolicyAndGuidance/DH_126666.

Di Sabatino, S., C. B. Hedquist, W. Carter, L. S. Leo, and S. J. H. Fernando (2009). Phoenix Urban Heat Island Experiment: Effects of Built Elements. *Conference Proceedings; American Meterological Soceity meeting*, J12.3.

Dong, Y., B. Forster, and C. Ticehurst (1997). Radar backscatter analysis for urban environments. *International Journal of Remote Sensing 18*(6), 1351–1364.

Donnay, J.-P. (1999). *Use of Remote Sensing Information in Planning*, Chapter 13, pp. 242–260. Advances in Spatial Science. Springer.

Dousset, B. (1989). AVHRR-derived cloudiness and surface temperature patterns over the los angeles area and their relationship to land use.

Dousset, B. and F. Gourmelon (2003). Satellite multi-sensor data analysis of urban surface temperatures and landcover. *ISPRS Journal of Photogrammetry and Remote Sensing 58*(1-2), 43–54.

Dousset, B., F. Gourmelon, K. Laaidi, A. Zeghnoun, E. Giraudet, P. Bretin, E. Mauri, and S. Vandentorren (2011). Satellite monitoring of summer heat waves in the Paris metropolitan area. *International Journal of Climatology 31*(2), 313–323.

DSS (2007). Dundee Satellite Receving Station, "Satellite Information." Accessed on July 23, 2012. http://www.sat.dundee.ac.uk/satellites.html.

Ebdon, D. (1985). *Statistics in geography. 2nd*. Basil Blackwell Ltd.

Eliasson, I. and M. Svensson (2003). Spatial air temperature variations and urban land use - a statistical approach. *Meteorological Applications 10*(2), 135–149.

ENVI (2012). Exelis Visual Information Solutions, "ENVI Modules." Accessed on August 13, 2012. http://www.exelisvis.com/productsservices/envi/envimodules.aspx.

EPSG (2012). European Geodetic Parameters Dataset, "EPSG Geodetic Parameters Dataset, Version 7.11.". Accessed August 16, 2012. http://www.epsg.org/geodetic.html.

Erbertseder, T., P. Tungalagsaikhan, M. Bittner, R. Meisner, M. Schroedter, and S. Dech (1999). Towards an operational atmospheric correction for AVHRR land surface products. *Geoscience and Remote Sensing Symposium, 1999. IGARSS '99 Proceedings. IEEE 1999 International 4*, 2227–2229 Vol.4.

Fish, P., G. Bennett, and P. Millard (1985). Heatwave morbidity and mortality in old age. *Age and ageing 14*(4), 243–245.

Forster, B. C. (1985). An examination of some problems and solutions in monitoring urban areas from satellite platforms. *International Journal of Remote Sensing 6*(1), 139–151.

Franklin, S. and P. Giles (1995). Radiometric processing of aerial and satellite remote-sensing imagery. *Computers & Geosciences 21*(3), 413–423.

Fung, W., K. Lam, J. Nichol, and M. Wong (2009). Derivation of nighttime urban air temperatures using a satellite thermal image. *Journal of Applied Meteorology and Climatology 48*(4), 863–872.

Gaffin, S. R., C. Rosenzweig, R. Khanbilvardi, L. Parshall, S. Mahani, H. Glickman, R. Goldberg, R. Blake, R. B. Slosberg, and D. Hillel (2008). Variations in New York city's urban heat island strength over time and space. *Theoretical and Applied Climatology 94*, 1–11.

Gallo, K., A. McNab, T. Karl, J. Brown, J. Hood, and J. Tarpley (1993). The use of NOAA AVHRR data for assessment of the urban heat island effect. *Journal of Applied Meteorology 32*(5), 899–908.

Gallo, K. P., J. D. Tarpley, A. L. McNab, and T. R. Karl (1995). Assessment of urban heat islands: a satellite perspective. *Atmospheric Research 37*(1-3), 37–43.

Gautier, L. (2011). "rpy2: A simple and efficient access to R from Python." Last modified June 2011. Accessed on August 18, 2012. `http://rpy.sourceforge.net/rpy2.html`.

GDAL (2012). *Geospatial Data Abstraction Library, Version 1.7.3*. *`http://www.gdal.org`*. Open Source Geospatial Foundation.

Geisser, S. (1993). *Predictive inference: An introduction*, Volume 55. Chapman & Hall.

Gentemann, C. L., C. J. Donlon, A. Stuart-Menteth, and F. J. Wentz (2003, February). Diurnal signals in satellite sea surface temperature measurements. *Geophys. Res. Lett. 30*(3).

Gillespie, A. (1985). Lithographic mapping of silicate rocks using tims. In *Proceedings TIMS Data User's Workshop*, Volume 86-38, Pasadena, CA, pp. 29–44. Jet Propulsion Laboratory: JPL Pub.

Gillespie, A., S. Rokugawa, S. Hook, T. Matsunaga, and A. Kahle (1999). Temperature/emissivity separation algorithm theoretical basis document, version 2.4. *ATBD contract NAS5-31372, NASA*.

Gillespie, A., S. Rokugawa, T. Matsunaga, J. Cothern, S. Hook, and A. Kahle (1998). A temperature and emissivity separation algorithm for advanced spaceborne thermal emission and reflection radiometer (aster) images. *Geoscience and Remote Sensing, IEEE Transactions on 36*(4), 1113–1126.

GLA (2006). Greater London Authority, "London's Urban Heat Island: A Summary for Decision Makers". Technical report, Greater London Authority (GLA).

Goetz, S., R. Halthore, F. Hall, and B. Markham (1995, DEC 20). Surface temperature retrieval in a temperate grassland with multiresolution sensors. *Journal of Geophysical Research-Atmospheres 100*(D12), 25397–25410.

Goïta, K. and A. Royer (1997). Surface temperature and emissivity separability over land surface from combined TIR and SWIR AVHRR data. *Geoscience and Remote Sensing, IEEE Transactions on 35*(3), 718–733.

Golden, J. (2004). The built environment induced urban heat island effect in rapidly urbanizing arid regions–a sustainable urban engineering complexity. *Environmental Sciences 1*(4), 321–349.

Goncalves Silva, A., R. De Alencar Lotufo, and R. Campos Machado (2001, oct). Toolbox of image processing for numerical python. In *Computer Graphics and Image Processing, 2001 Proceedings of XIV Brazilian Symposium on*, pp. 402.

Hadjimitsis, D. G., C. R. I. Clayton, and V. S. Hope (2004). An assessment of the effectiveness of atmospheric correction algorithms through the remote sensing of some reservoirs. *International Journal of Remote Sensing 25*(18), 3651–3674.

Hajat, S., S. Sheridan, M. Allen, M. Pascal, K. Laaidi, A. Yagouti, U. Bickis, A. Tobias, D. Bourque, B. Armstrong, et al. (2010). Heat-health warning systems: A comparison of the predictive capacity of different approaches to identifying dangerously hot days. *American Journal of Public Health 100*(6), 1137.

Hammond, R. and P. S. McCullagh (1978). *Quantitative Techniques in Geography*. Oxford University Press.

Hansen, J., M. Sato, and R. Ruedy (2012). Perception of climate change. *Proceedings of the National Academy of Sciences*.

Hardy, A., N. Forsythe, T. Holderness, and H. J. Fowler (2012). Development of a snow covered area (SCA) product for the Western Himalaya using AVHRR imagery, [In review].

Harlan, S., A. Brazel, L. Prashad, W. Stefanov, and L. Larsen (2006). Neighborhood microclimates and vulnerability to heat stress. *Social Science & Medicine 63*(11), 2847–2863.

Hedquist, B. C., J. A. Brazel, S. Di Sabatino, W. Carter, and S. J. H. Fernando (2009). Phoenix Urban Heat Island Experiment: Micrometeorological Aspects. *Conference Proceedings; American Meterological Soceity meeting*, J12.2.

Houghton, J. et al. (2001). *Climate change 2001: the scientific basis*, Volume 881. Cambridge University Press Cambridge, UK.

Howard, L. (1833). *The climate of London: deduced from meteorological observations made in the metropolis and at various places around it*. Harvey and Denton.

IPCC (2007, November). Climate Change 2007 Synthesis Report. Synthesis Report Four, The International Panel on Climate Change.

Jaggi, S., D. Quattrochi, and R. Baskin (1992). An algorithm for the estimation of bounds on the emissivity and temperatures from thermal multispectral airborne remotely sensed data. In *JPL, Summaries of the Third Annual JPL Airborne Geoscience Workshop.*, Volume 2.

Jenerette, G., S. Harlan, A. Brazel, N. Jones, L. Larsen, and W. Stefanov (2007). Regional relationships between surface temperature, vegetation, and human settlement in a rapidly urbanizing ecosystem. *Landscape Ecology 22*(3), 353–365.

Jimenez-Munoz, J.-C. and J. A. Sobrino (2008, OCT). Split-window Coefficients for Land Surface Temperature Retrieval from low-resolution thermal infrared sensors. *IEEE Geoscience and Remote Sensing Letters 5*(4), 806–809.

Jin, M., R. E. Dickinson, and D. Zhang (2005, May). The Footprint of Urban Areas on Global Climate as Characterized by modis. *J. Climate 18*(10), 1551–1565.

Johnson, H., R. S. Kovats, G. McGregor, J. Stedman, M. Gibbs, H. Walton, L. Cook, and E. Black (2005). The impact of the 2003 heat wave on mortality and hospital admissions in england. *Health Statistics Quarterly 25*, 6–11.

Jones, E., T. Oliphant, P. Peterson, et al. (2001). SciPy: Open source scientific tools for Python. `http://www.scipy.org/`.

Jones, P., D. Lister, and Q. Li (2008). Urbanization effects in large-scale temperature records, with an emphasis on China. *J. Geophys. Res 113*, D16122.

Jones, P. D. and D. H. Lister (2009). The urban heat island in Central London and urban-related warming trends in central London since 1900. *Weather 64*(12), 323–327.

Kalluri, S. N. V. and R. O. Dubayah (1995). Comparison of atmospheric correction models for thermal bands of the advanced very high resolution radiometer over fife. *Journal of Geophysical Research 100*, 25411–25418.

Kato, S. and Y. Yamaguchi (2005). Analysis of urban heat-island effect using ASTER and ETM+ data: Separation of anthropogenic heat discharge and natural heat radiation from sensible heat flux. *Remote Sensing of Environment 99*(1-2), 44–54.

Kato, S. and Y. Yamaguchi (2007). Estimation of storage heat flux in an urban area using ASTER data. *Remote Sensing of Environment 110*(1), 1–17.

Kawashima, S., T. Ishida, M. Minomura, and T. Miwa (2000, September). Relations between Surface Temperature and Air Temperature on a Local scale during winter nights. *J. Appl. Meteor. 39*(9), 1570–1579.

Kealy, P. and A. Gabell (1990). Estimation of emissivity and temperature using alpha coefficients. In *Proc. 2nd TIMS Workshop*, pp. 90–55.

Khorram, S., J. Brockhaus, and H. Cheshire (1987, March). Comparison of Landsat MSS and TM Data for Urban Land-Use Classification. *Geoscience and Remote Sensing, IEEE Transactions on GE-25*(2), 238–243.

Kim, Y.-H. and J.-J. Baik (2002). Maximum Urban Heat Island Intensity in Seoul. *Journal of Applied Meteorology 41*, 651–659.

Knight, S., C. Smith, and M. Roberts (2010). Mapping Manchester's urban heat island. *Weather 65*(7), 188–193.

Kolokotroni, M. and R. Giridharan (2008). Urban heat island intensity in London: An investigation of the impact of physical characteristics on changes in outdoor air temperature during summer. *Solar Energy 82*(11), 986–998.

Koppe, C., S. Kovats, G. Jendritzky, B. Menne, et al. (2004). Heat-waves: risks and responses. *Health and Global Environmental Change Series 2*.

Kovats, R. and S. Hajat (2008). Heat stress and public health: a critical review. *Annu. Rev. Public Health 29*, 41–55.

Kovats, R. S. and L. K. Ebi (2006). Heatwaves and public health in Europe. *The European Journal of Public Health 16*(6), 592–599.

Kovats, R. S., H. Johnson, and C. Griffiths (2006). Mortality in southern England during the 2003 heat wave by place of deaths. *Health Statistics Quarterly 29*, 6–8.

Latifovic, R., A. P. Trishchenko, J. Chen, W. B. Park, K. V. Khlopenkov, R. Fernandes, D. Pouliot, C. Ungureanu, Y. Luo, S. Wang, A. Davidson, and J. Cihlar (2005). Generating historical AVHRR 1 km baseline satellite data records over canada suitable for climate change studies. *Canadian Journal of Remote Sensing 31*(5), 324–346.

Lee, D. (1992). Urban warming? An analysis of recent trends in London's heat island. *Weather 47*(2), 50–55.

Lee, H.-Y. (1993). An application of NOAA AVHRR thermal data to the study of urban heat islands. *Atmospheric Environment. Part B. Urban Atmosphere 27*(1), 1–13.

Lejdfors, C. and L. Ohlsson (2005). Pygpu: A high-level language for high-speed image processing. *Rapport Technique, Université de Lund, Suede*.

Liang, B. and Q. Weng (2008, SEP). Multiscale analysis of census-based land surface temperature variations and determinants in Indianapolis, United States. *Journal of Urban Planning and Development 134*(3), 129–139.

Lillesand, M., W. Kiefer, and W. Chipman (2004). *Remote Sensing and Image Interpretation* (Fifth ed.). John Wiley and Sons.

Lo, C. (2003). *Remotely Sensed Cities*, Chapter 7 Zone-based estimation of population and housing units from satellite-generated land use/land cover maps, pp. 157–180. Taylor & Francis.

Lo, C. and D. Quattrochi (2003). Land-use and land-cover change, urban heat island phenomenon, and health implications: a remote sensing approach. *Photogrammetric engineering and remote sensing 69*(9), 1053–1063.

Lu, D., P. Mausel, E. Brondizio, and E. Moran (2002). Assessment of atmospheric correction methods for landsat tm data applicable to amazon basin lba research. *International Journal of Remote Sensing 23*(13), 2651–2671.

Lu, D. and Q. Weng (2006). Spectral mixture analysis of ASTER images for examining the relationship between urban thermal features and biophysical descriptors in Indianapolis, Indiana, USA. *Remote Sensing of Environment 104*(2), 157–167.

Martelli, A. (2006). *Python in a Nutshell*. O'Reilly Media, Inc.

Mather, P. M. (2004). *Computer Processing of Remotely-Sensed Images*. John Wiley & Sons, Ltd.

Matson, M., E. McClain, D. McGinnis, and J. Pritchard (1978). Satellite detection of urban heat islands. *Monthly Weather Review 106*, 1725.

McCarthy, J. (2001). *Climate change 2001: impacts, adaptation, and vulnerability: contribution of Working Group II to the third assessment report of the Intergovernmental Panel on Climate Change*. Cambridge Univ Press.

McClain, E., W. Pichel, C. Walton, Z. Ahmad, and J. Sutton (1983). Multichannel improvements to satellite-derived global sea surface temperatures. *Advances in Space Research 2*(6), 43–47.

McKinney, W. (2011, November). pandas: a foundational python library for data analysis and statistics. In *Python for High Performance and Scientific Computing*.

Mehrotra, S., C. Natenzon, A. Omojola, R. Folorunsho, J. Gilbride, and C. Rosenzweig (2009). Framework for city climate risk assessment. In *Fifth Urban Research Symposium, Marseille, France*.

Mehrotra, S., C. Rosenzweig, W. Solecki, C. Natenzon, , A. Omojola, R. Folorunsho, and J. Gilbride (2011). *Cities, disasters, and climate risk. Climate Change and Cities: First Assessment Report of the Urban Climate Change Research Network*, Chapter 2, pp. 15–42. Cambridge University Press.

Memon, R. A., D. Y. Leung, and C.-H. Liu (2009). An investigation of urban heat island intensity (UHII) as an indicator of urban heating. *Atmospheric Research 94*(3), 491–500.

Menut, L., C. Flamant, J. Pelon, and P. Flamant (1999). Urban boundary-layer height determination from lidar measurements over the Paris area. *Applied Optics 38*(6), 945–954.

Mesev, V. (Ed.) (2003). *Remotely Sensed Cities*. Taylor & Francis.

Miller, R. B. and C. Small (2003). Cities from space: potential applications of remote sensing in urban environmental research and policy. *Environmental Science & Policy 6*(2), 129–137.

Millman, K. J. and M. Aivazis (2011). Python for scientists and engineers. *Computing in Science Engineering 13*(2), 9–12.

Nagol, J. R., E. F. Vermote, and S. D. Prince (2009). Effects of atmospheric variation on AVHRR NDVI data. *Remote Sensing of Environment 113*(2), 392–397.

Nichol, J. (1994). A GIS-based approach to microclimate monitoring in Singapore's high-rise housing estates. *Photogrammetric Engineering and Remote Sensing 60*(10), 1225–1232.

Nichol, J. (2003). *"GIS and remote sensing in urban heat islands in the Third World."*, Chapter 11 in Remotely Sensed Cities, 243. (Taylor & Francis).

Nichol, J. (2005). Remote sensing of urban heat islands by day and night. *Photogrammetric Engineering and Remote Sensing 71*(5), 613–622.

Nichol, J. (2009). An emissivity modulation method for spatial enhancement of thermal satellite images in urban heat island analysis. *Photogrammetric Engineering and Remote Sensing 75*(5), 547–556.

Nichol, J. E. (1996). Analysis of the urban thermal environment with LANDSAT data. *Environment and Planning B: Planning and Design 23*, 733–747.

Nichol, J. E., W. Y. Fung, K. se Lam, and M. S. Wong (2009). Urban heat island diagnosis using aster satellite images and 'in situ' air temperature. *Atmospheric Research 94*(2), 276–284.

NOAA (2009). National Ocanagaphic and Atmospheric Administration, "NOAA KLM User's Guide." Accessed on June 4, 2009. `http://www.ncdc.noaa.gov/oa/pod-guide/ncdc/docs/klm/cover.htm`.

NOAA (2012). National Ocanagaphic and Atmospheric Administration, "Revised post-launch calibration of channels 1 and 2 of the Advanced Very High Resolution Radiometer on board the NOAA-14 spacecraft." accessed on june 16, 2012. `http://noaasis.noaa.gov/noaasis/ml/aboutn14vis.html`.

Oke, T. (1976). The distinction between canopy and boundary-layer urban heat islands. *Atmosphere 14*(4), 268–277.

Oke, T. R. (1973). City size and the urban heat island. *Atmospheric Environment 7*, 769–779.

Oke, T. R. (1981). Canyon Geometry and the Nocturnal Urban Heat-Island - Comparison of scale model and field observations. *The Journal of Climatology 1*(3), 237–254.

Oke, T. R. (1982). The Energetic Basis of the Urban Heat-Island. *Quarterly Journal of the Royal Meterological Society 108*(455), 1–24.

Oke, T. R. (1987). *Boundary Layer Climates*. Methuen.

Oliphant, T. (2007). *Multidimensional Iterators in NumPy*, Chapter 27 in Beautiful Code. (O'Reilly).

Owen, T. W., T. N. Carlson, and R. R. Gillies (1998). An assessment of satellite remotely-sensed land cover parameters in quantitatively describing the climatic effect of urbanization. *International Journal of Remote Sensing 19*(9), 1663–1681.

Pérez, F., B. Granger, and J. Hunter (2011). Python: An ecosystem for scientific computing. *Computing in Science Engineering 13*(2), 13 –21.

Pongracz, R., J. Bartholy, and Z. Dezso (2006). Remotely sensed thermal information applied to urban climate analysis. *Advances in Space Research 37*(12), 2191–2196.

Price, J. (1984). Land surface temperature measurements from the split window channels of the NOAA 7 Advanced Very High Resolution Radiometer. *Journal of Geophysical Research 89*(D5), 7231–7237.

Price, J. C. (1987). Radiometric calibration of satellite sensors in the visible and near infrared: History and outlook. *Remote Sensing of Environment 22*(1), 3 – 9.

Prihodko, L. and S. Goward (1997). Estimation of air temperature from remotely sensed surface observations. *Remote Sensing of Environment 60*(3), 335–346.

QGIS (2012). Quantum GIS Development Team, "Quantum GIS Geographic Information System." Accessed on August 19, 2012. `http://qgis.osgeo.org`.

Rahman, H. and G. Dedieu (1994). SMAC: a simplified method for the atmospheric correction of satellite measurements in the solar spectrum. *International Journal of Remote Sensing 15*(1), 123–143.

Ren, G. Y., Z. Y. Chu, Z. H. Chen, and Y. Y. Ren (2007, March). Implications of temporal change in urban heat island intensity observed at Beijing and Wuhan stations. *Geophysical Research Letters 34*(5).

Riverbank Computing Limited (2012). What is PyQt? Accessed on August 21 2012. http://www.riverbankcomputing.com/software/pyqt/intro.

Robinson, P. J. (2001). On the definition of a heat wave. *Journal of Applied Meteorology 40*(4), 762–775.

Rosenzweig, C. (2011, September). All climate is local. *Scientific American 305*, 70–73.

Rosenzweig, C., W. Solecki, S. Hammer, and S. Mehrotra (2011). *Urban Climate Change in Context. Climate Change and Cities: First Assessment Report of the Urban Climate Change Research Network*, Chapter 1, pp. 3–11. Cambridge Univ Press.

Rosenzweig, C., W. D. Solecki, L. Parshall, M. Chopping, G. Pope, and R. Goldberg (2005). Characterizing the urban heat island in current and future climates in New Jersey. *Global Environmental Change Part B: Environmental Hazards 6*(1), 51–62.

Roth, M., T. R. Oke, and W. J. Emery (1989). Satellite-derived urban heat islands from three coastal cities and the utilization of such data in urban climatology. *International Journal of Remote Sensing 10*(11), 1699–1720.

Schädlich, S., F. M. Göottsche, and F. S. Olesen (2001). Influence of Land Surface Parameters and Atmosphere on METEOSAT Brightness temperatures and generation of land surface temperature maps by temporally and spatially interpolating atmospheric correction. *Remote Sensing of Environment 75*(1), 39–46.

Schmugge, T. J. and G. M. Schmidt (1998). Surface Temperature Observations from AVHRR in FIFE. *Journal of the Atmospheric Sciences*, 1239–1246.

Shepherd, J. M., H. Pierce, and A. J. Negri (2002). Rainfall Modification by Major Urban Areas: Observations from Spaceborne rain radar on the TRMM satellite. *Journal of Applied Meteorology 41*, 689–701.

Snyder, W., Z. Wan, Y. Zhang, and Y. Feng (1998). Classification-based emissivity for land surface temperature measurement from space. *International Journal of Remote Sensing 19*(14), 2753–2774.

Sobrino, J., C. Coll, and V. Caselles (1991). Atmospheric correction for land surface temperature using NOAA-11 avhrr channels 4 and 5. *Remote Sensing of Environment 38*(1), 19–34.

Sobrino, J., J. Jimenez-Muoz, G. Soria, M. Romaguera, L. Guanter, J. Moreno, A. Plaza, and P. Martinez (2008, Feb.). Land Surface Emissivity Retrieval From Different VNIR and TIR Sensors. *IEEE Transactions on Geoscience and Remote Sensing 46*(2), 316–327.

Solecki, W. D., C. Rosenzweig, L. Parshall, G. Pope, M. Clark, J. Cox, and M. Wiencke (2005). Mitigation of the heat island effect in urban New Jersey. *Global Environmental Change Part B: Environmental Hazards 6*(1), 39–49.

Stabel, E. and P. Fischer (2001). Satellite radar interferometric products for the urban application domain. *Advances in Environmental Research 5*(4), 425–433.

Stathopoulou, M. and C. Cartalis (2009). Downscaling AVHRR land surface temperatures for improved surface urban heat island intensity estimation. *Remote Sensing of Environment 113*(12), 2592–2605.

Stephens, A., P. James, D. Alderson, S. Pascoe, S. Abele, A. Iwi, and P. Chiu (2012). The challenges of developing an open source, standards-based technology stack to deliver the latest UK climate projections. *International Journal of Digital Earth 5*(1), 43–62.

Streutker, D. (2002). A remote sensing study of the urban heat island of Houston, Texas. *International Journal of Remote Sensing 23*(13), 2595–2608.

Streutker, D. R. (2003). Satellite-measured growth of the urban heat island of Houston, Texas. *Remote Sensing of Environment 85*(3), 282–289.

Sutton, T. (2010). Announcing the release of QGIS 1.6 'Copiapó', Accessed on August 20, 2012. `http://blog.qgis.org/node/146`.

Teillet, P. (1992). An algorithm for the radiometric and atmospheric correction of AVHRR data in the solar reflective channels. *Remote Sensing of Environment 41*(2-3), 185–195.

Terrel, A. (2011). From equations to code: Automated scientific computing. *Computing in Science Engineering 13*(2), 78 –82.

To, P. H., J. Nichol, and R. Tse (2011). Temporal characteristics of thermal satellite images for urban climate study. In *Urban Remote Sensing Event (JURSE)*, pp. 137–140.

Tobler, W. (1969). Satellite confirmation of settlement size coefficients. *Area*, 30–34.

Tomlinson, C. J., L. Chapman, J. E. Thornes, and C. J. Baker (2012). Derivation of Birmingham's summer surface urban heat island from MODIS satellite images. *International Journal of Climatology 32*, 214–224.

Toolbox, T. E. (2012). "Solids - Specific Heats." Accessed on August 30, 2012. http://www.engineeringtoolbox.com/specific-heat-solids-d_154.html.

Tran, H., D. Uchihama, S. Ochi, and Y. Yasuoka (2006). Assessment with satellite data of the urban heat island effects in Asian mega cities. *International Journal of Applied Earth Observation and Geoinformation 8*(1), 34–48.

Tungalagsaikhan, P. and K. P. Guenther (2007). "NOAA AVHRR derived Land Surface Temperature Maps (LST)." Last updated December, 2007. http://eoweb.dlr.de/short_guide/d-lst.html.

UKMO (2011). United Kingdom Met Office, "Heat-Health Watch." Accessed on October 5, 2011. http://www.metoffice.gov.uk/weather/uk/heathealth/.

UKMO (2012). United Kingdom Met Office, "MIDAS Land Surface Stations data (1853-current), NCAS British Atmospheric Data Centre, 2006-2012." http://badc.nerc.ac.uk/view/badc.nerc.ac.uk__atom__dataent_ukmo-midas.

UN (2009). United Nations,"World Urbanization Prospects, the 2009 Revision: Highlights." Accessed on May 31, 2011. http://esa.un.org/unpd/wup/doc_highlights.htm.

Unwin, D. J. (1980). The Synoptic Climatology of Birmingham's Urban Heat Island, 1965-84. *Weather 35*(2), 43–50.

Valor, E. and V. Caselles (1996). Mapping land surface emissivity from NDVI: Application to European, african, and south american areas. *Remote Sensing of Environment 57*(3), 167–184.

Van De Griend, A. A. and M. Owe (1993). On the relationship between thermal emissivity and the normalized difference vegetation index for natural surfaces. *International Journal of Remote Sensing 14*(6), 1119–1131.

van der Linden, S. and P. Hostert (2009). The influence of urban structures on impervious surface maps from airborne hyperspectral data. *Remote Sensing of Environment 113*(11), 2298–2305.

van der Walt, S., S. Colbert, and G. Varoquaux (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering 13*(2), 22–30.

Varrazzo, D. (2012). "Psycopg." Accessed on August 18, 2012.http://initd.org/psycopg/.

Vogt, J., A. Viau, and F. Paquet (1997). Mapping regional air temperature fields using satellite-derived surface skin temperatures. *International Journal of Climatology 17*(14), 1559–1579.

Voogt, J. A. and T. R. Oke (2003). Thermal remote sensing of urban climates. *Remote Sensing of Environment 86*(3), 370–384.

Watson, K. (1992). Spectral ratio method for measuring emissivity. *Remote Sensing of Environment 42*(2), 113–116.

Weng, Q., D. Lu, and J. Schubring (2004). Estimation of land surface temperature-vegetation abundance relationship for urban heat island studies. *Remote Sensing of Environment 89*(4), 467–483.

Wilby, R. L. (2003). Past and projected trends in London's urban heat island. *Weather 58*(7), 251–260.

Wilby, R. L. (2007). A review of climate change impacts on the built environment. *Built Environment Journal 33*, 31–45.

Wilby, R. L. (2008). Constructing climate change scenarios of urban heat island intensity and air quality. *Environment and Planning B: Planning and Design 35*, 902–919.

Wilby, R. L., P. D. Jones, and D. H. Lister (2011). Decadal variations in the nocturnal heat island of London. *Weather 66*(3), 59–64.

Worboys, M. and M. Duckham (2004). *GIS: A computing perspective*. CRC Press.

Xian, G., M. Crane, and C. McMahon (2008, APR). Quantifying multi-temporal urban development characteristics in Las vegas from landsat and aster data. *Photogrammetric Engineering and Remote Sensing 74*(4), 473–481.

Xu, W., M. Wooster, and C. Grimmond (2008). Modelling of urban sensible heat flux at multiple spatial scales: A demonstration using airborne hyperspectral imagery of Shanghai and a temperature-emissivity separation approach. *Remote Sensing of Environment 112*(9), 3493–3510.

Yu, Y., J. L. Privette, and A. C. Pinheiro (2008). Evaluation of split-window land surface temperature algorithms for generating climate data records. *IEEE Transactions on Geoscience and Remote Sensing 46*(1), 179–192.

Appendix A

Software to read NOAA level 1B AVHRR data

Prior to the provision of pre-processed AVHRR data in GeoTiff format by Dundee Satellite Receiving Station (DSS), two software modules were written to read AVHRR data and apply a radiometric correction to the thermal channels four and five. Both modules were based on the DSS n1bx software written in the C programming language. The scanline module reads NOAA level 1B 10-bit packed binary files and coverts them to ASCII images in the PGM format. The radiance module uses the output from scanline to apply the radiometric calibration to themal bands four and five as defined in NOAA (2009) and returns corrected images in the PGM format. Figure A.1 shows an example output AVHRR band four scene which has been extracted using the scanline module, and then radiometrically corrected using the radiance module.

A.1 Scanline module

The scanline module unpacks and reads NOAA level 1B AVHRR. For brevity only the "readscanline" and "scanlinecal" functions are listed below. The readscanline function unpacks and reads the raw data for each scanline before

performing a big to little endian conversion and writing the pixel values of each AVHRR band to a separate text file. The scanlinecal function repeats the process for the scan line radiometric calibration coefficients for thermal bands 4 and 5.

A.1.1 Read scan line function “readscanline”

```
1  /*
2  readscanline - function to read and uncompress NOAA L1B 10-bit packed data.
3  */
4  void readscanline(FILE *infile, FILE *outfile_ch1, FILE *outfile_ch2, FILE *outfile_ch3,
       FILE *outfile_ch4, FILE *outfile_ch5)
5  {
6          ///arrays for 10-bit packed DN values (3 pixels (30bits) per integer (32bits)).
7          unsigned int top[3414],mid[3414],end[3414];
8          int i=0; ///counter
9
10         ///loop along scanline
11         for (i=0;i<3414;i++)
12                 {
13                 unsigned char charray[4];
14                 ///array to read one integer (four octets = four chars)
15                 int j = 0; ///counter
16
17                 ///read in four chars.
18                 for(j=0;j<4;j++)
19                         {
20                         fscanf(infile,"%c",&charray[j]);
21                         }
22
23                 ///convert 4 chars to one integer (Endian conversion).
24                 unsigned int val;
25                 val = (charray[0] << 24) + (charray[1] << 16) + (charray[2] << 8) + (
                       charray[3] << 0);
26
27                 ///Start binary counting of integers.
28                 int k= 0;                       ///counter
29                 unsigned int b[NUM_OF_BITS];    ///array to hold binary representation.
30                 int n = NUM_OF_BITS;            ///shorthand for 32.
31                 int mask = 1 << (n - 1);        ///define mask.
32
33                 ///start binary count loop
34                 for (k=0; k < n; k++)
35                         {
36                         if ((val & mask) == 0)
37                                 b[k] = 0;       ///place 0's if mask is 0
38                         else
```

```
                                b[k] = 1;       ///else place 1.
                                val <<= 1;      ///shift integer by one bit.
                        }
                ///Extract each 10-bit pixel.
                unsigned int c1 = 0, c2 = 0, c3 = 0;  ///ints to hold each 10bit pixel
                    value.
                for (k=2;k<12;k++)      ///first 10bit pixel
                        {
                        ///multiply binary value (1 or 0) by its position in row to give
                             numeric form (e.g. 32)
                        b[k]= b[k] * (pow(2, (11-k)));
                        c1 += b[k];     ///sum binary form values.
                        }
                for (k=12;k<22;k++)     ///second 10bit pixel
                        {
                                b[k]= b[k] * (pow(2, (21-k)));
                                c2 += b[k];
                        }

                for (k=22;k<32;k++)     ///third 10bit pixel
                        {
                                b[k]= b[k] * (pow(2, (31-k)));
                                c3 += b[k];
                        }
                        ///Pass first, second and third pixels to arrays
                        top[i] = c1;
                        mid[i] = c2;
                        end[i]=c3;
                }
/// Now write the values for each of the bands to a file.
for (i=0;i<3410;i+=5)
        {
        fprintf(outfile_ch1, "%d %d %d ",top[i],end[i+1],mid[i+3]);
        fprintf(outfile_ch2, "%d %d %d ",mid[i],top[i+2],end[i+3]);
        fprintf(outfile_ch3, "%d %d %d ",end[i],mid[i+2],top[i+4]);
        fprintf(outfile_ch4, "%d %d %d ",top[i+1],end[i+2],mid[i+4]);
        fprintf(outfile_ch5, "%d %d %d ",mid[i+1],top[i+3],end[i+4]);
        }
        ///compensate for 10240/3 = 3413.333...add on last two values.
        fprintf(outfile_ch1, "%d %d",top[3410],end[3411]);
        fprintf(outfile_ch2, "%d %d",mid[3410],top[3412]);
        fprintf(outfile_ch3, "%d %d",end[3410],mid[3412]);
        fprintf(outfile_ch4, "%d %d",top[3411],end[3412]);
        fprintf(outfile_ch5, "%d %d",mid[3411],top[3413]);
///print newline at end of scanline.
fprintf(outfile_ch1,"\n");
fprintf(outfile_ch2,"\n");
fprintf(outfile_ch3,"\n");
fprintf(outfile_ch4,"\n");
fprintf(outfile_ch5,"\n"); }
```

A.1.2 Read thermal scan line radiometric calibration coefficients function “scanlinecal”

```
1  void scanlinecal (FILE *infile, FILE *outfile)
2  {
3  int t = 0;
4  int i = 0;
5  unsigned char charray[4];
6  int val = 0;
7  double cf[3];
8
9  for (i = 0;i<3;i++)
10 {
11         for (t=0;t<4;t++)
12         {               fscanf(infile, "%c",&charray[t]);
13         }
14         val = (charray[0] << 24) + (charray[1] << 16) + (charray[2] << 8) + (charray[3]
               << 0);
15         double valve = 0;       valve = val / (pow(10,6));
16         fprintf(outfile,"%lf ",valve);
17 }
18 ///print newline after calibration values for each scanline
19 fprintf(outfile,"\n");
20 }
```

A.2 Radiance module

The radiance module reads the unpacked data from the scanline module and applies the radiometric correction to AVHRR thermal bands four and five as defined by NOAA (2009).

```
1  /** radiance.c T.Holderness June 2009
2  *calibrate AVHRR KLM thermal bands to Earth Scene Radiance
3  *Requires scene data dn calibration coefficients from scanline.c **/
4
5  ///Formula for computing Earth Scene Radiance from NOAA KLM POD Guide ///NE = a0 + a1.DN
       + a2.DN^2
6  #include<stdio.h>
7  #include<stdlib.h>
8  #include<math.h>
9
10 int main(void)
11 { ///file IO
12 FILE *data, *cf, *outfile;
13 data = fopen("/home/holderness/dump/img/img.n1b.ch4","r"); ///from scanline.c
14 cf = fopen("/home/holderness/dump/img/img_ch4cal.asc","r");///from scanline.c
```

```
15 outfile = fopen("/home/holderness/dump/img/img.esr.ch4","w");
16
17 int i = 0, t = 0;                ///counters
18 unsigned int DN[2048];           ///Scan line DN values unsigned
19 int DN_ESR[2048];                ///DN Earth Scene Radiance
20 char varc;                       ///PGM magic letter
21 int a,b,c,d;                     ///PGM header values
22 double c1[5506],c2[5506],c3[5506];   ///arrays for each scan-line calibration value
23 ///strip PGM header from input image.
24 fscanf(data,"%c%d %d %d %d",&varc ,&a,&b,&c,&d);
25 ///write PGM header to New Earth Scene Radiance File
26 fprintf(outfile,"%c%d %d %d %d\n",varc,a,b,c,d);
27 ///read one image (consisting of 5506 scanlines)
28 int f;                                  ///counter
29 for (f=0;f<5506;f++)
30 {
31         fscanf(cf, "%lf %lf %lf",&c1[f],&c2[f],&c3[f]);
32         fprintf(stderr, "%lf %lf %lf\n",c1[f],c2[f],c3[f]);
33         ///read one scanline
34         for (i=0;i<2048;i++)
35         {
36                 fscanf(data,"%d",&DN[i]);
37                 ///calculate Earth Scene Radiance
38                 DN_ESR[i] = (c1[f] + (c2[f]*DN[i]) + (c3[f] * (pow(DN[i],2))));
39                 if (DN_ESR[i] < 0){
40                         DN_ESR[i] = 1;
41                         }
42                 fprintf(outfile,"%d ",DN_ESR[i]);
43         }
44         ///print newline after scan line
45         fprintf(outfile,"\n"); }
46 fclose(data);
47 fclose(cf);
48 fclose(outfile);
49 return 0;
50 }
```

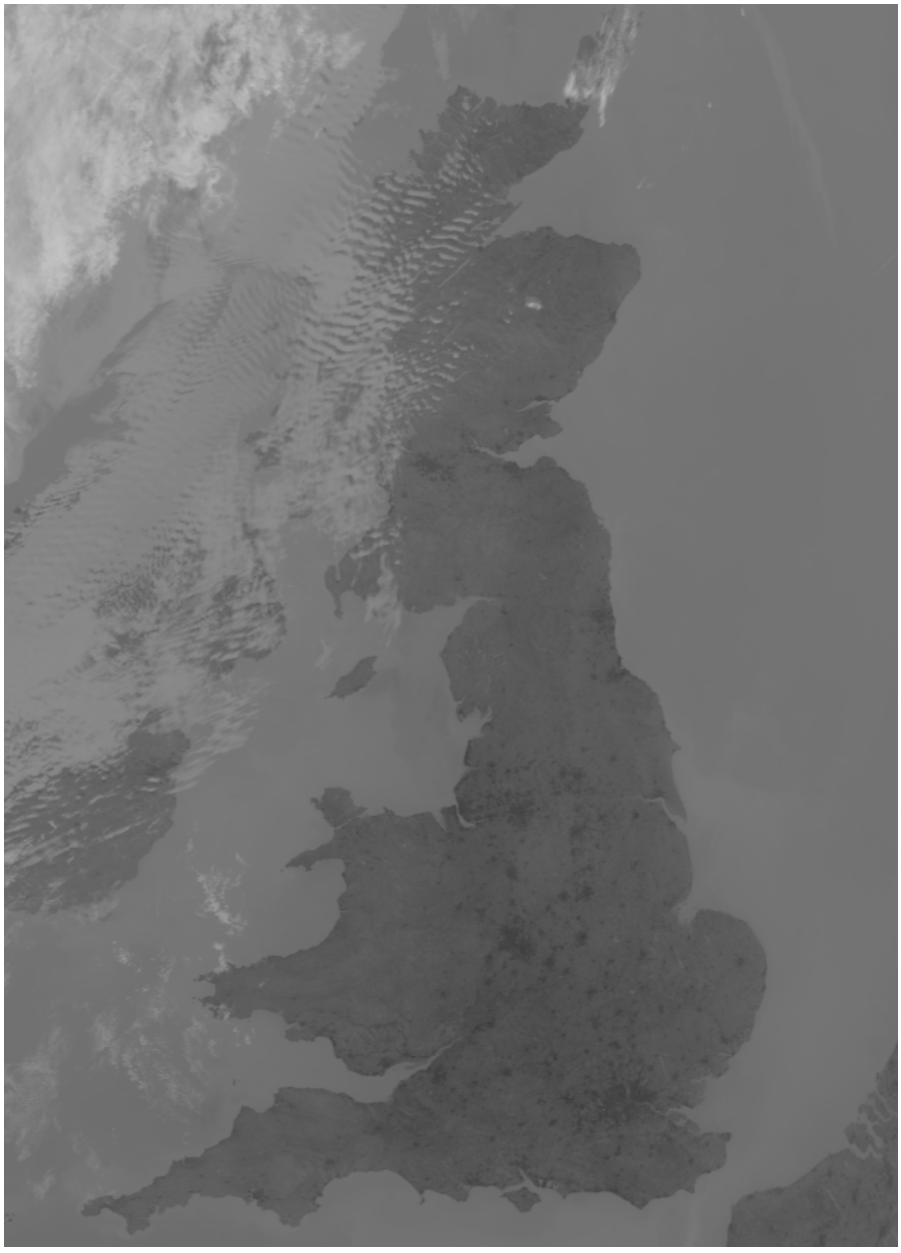


Figure A.1: Example AVHRR band four image of the British Isles after extraction from NOAA level 1B 10-bit packed format and calibrated to Earth scene radiance. Contrast stretching has not been applied. Dark areas are warmer, with major urban conurbations clearly visible.

Appendix B

Software to process raster imagery in Python

To facilitate pre-processing and analysis of AVHRR GeoTiffs a suite of Python tools for processing geospatial raster data was developed. Three key modules as part of the PyRaster suite were developed:

1. RasterIO - a library of functions to convert geospatial raster formats to/from Numpy masked arrays

2. PyAVHRR - a library of sensor specific functions for processing AVHRR data

3. Raster Processing Suite - A plugin for the Quantum GIS package for exploratory analysis of raster data

The API documentation for the RasterIO and PyAVHRR modules and the associated source code is shown below. The source code for the Raster Processing Suite module is also shown, although for brevity ancillary files to load the plugin in QGIS and the PyQt GUI XML schema are not included.

B.1 Documentation for RasterIO module

RasterIO - Library of functions to convert geospatial raster formats to/from Numpy masked arrays.

Introduction

This library contains wrapper functions for GDAL Python bindings, converting data to Numerical Python multi-dimensional arrays in memory for processing. Subsequent generated arrays can be written to disk in the standard geospatial GeoTiff format.

Notes

- Error checking - RasterIO contains minimal user-level error checking
- In the source code a GDAL pointer to a raster file is termed 'dataset'

Supported Formats

- Input:
  - RasterIO supports reading any GDAL supported raster format
- Output:
  - RasterIO generates GeoTiff files by default (this can be modified in the source code)
  - GeoTiffs are created with embedded binary header files containing geo information

Supported Datatypes

- Raster IO supports Float32 and Int16 data types
- The default datatype is Float32
- Boolean datasets use Int16 datatypes

NoDataValue

If the input data has no recognisable NoDataValue (readable by GDAL) then the input NoDataValue is assumed to be 9999. This can be changed by manu-

ally specifying an input NoDataVal when calling readrasterbands(). In accordance with GDAL the output data NoDataValue is 9999 or 9999.0 or can be manually set by when writrasterbands(). When using unsigned integer data types the default output NoDataValue will be 0.

How to use documentation

Documentation for module functions is provided as Python docstrings, accessible from an interactive Python terminal. Within docstrings, examples from an interactive Python console are identified using '>>>' Further information is given to developers within the source code using '#' comment strings. To view this text and a list of available functions call the Python in-built help command, specifying module name

```
>>> import rasterIO
>>> help(rasterIO)
...this text...
```

For help on a specific function call the Python in-built help command, specifying module.function

```
>>> import rasterIO
>>> help(rasterIO.wkt2epsg)
Help on function wkt2epsg in module rasterIO
wkt2epsg(wkt)
Accepts well known text of Projection/Coordinate
Reference System and generates EPSG code
```

How to access functions

To access functions, import the module to Python and call the desired function, assigning the output to a named variable. Note that the primary input datatype (default) for all functions is either a Numpy array or a Numpy masked array. Use the rasterIO module to convert Numpy arrays to/from Geospatial raster data formats, for example to read a raster:

```
>>> import rasterIO as rio
```

```
>>> pointer = rio.opengdalraster('file.tif')
>>> band_number = 1
>>> b1_data = rio.readrasterband(pointer, band_number)
```

Optional function arguments are shown in document strings in brackets.

Dependencies

Python 2.5 or greater Numerical python (Numpy) 1.2.1 or greater (1.4.1 recommended)

- Note that due to bugs in Numpy.ma module, Numpy 1.4.1 or greater is required to support masked arrays of integer values. See comments in reasrasterband() for more information.

License & Authors

Copyright: Tom Holderness & Newcastle University

Released under the Simplified BSD License (see license.txt)

Version: 1.1.1

B.1.1 Functions

opengdalraster(*filename*)

Accepts filename for GDAL compatible file and returns a GDAL pointer (dataset).

readrastermeta(*dataset*)

Accepts GDAL raster dataset (pointer) and returns, gdal_driver, XSize, YSize, projection info(well known text), geotranslation data.

readrasterband(*dataset*, *aband*, *NoDataVal*=`None`, *masked*=`True`)

Accepts GDAL raster dataset (pointer) and band number, returns Numpy array.

newgdalraster(*outfile*, *format*, *XSize*, *YSize*, *geotrans*, *epsg*, *num_bands*, *gdal_dtype*)

Accepts filename, format, XSize, YSize, geotransformation, epsg, number_of_bands, gdal_datatype and returns gdal pointer to new file.

This is a lower level function that allows users to control data output stream directly, use for specialist cases such as varying band data types or memory limited read-write situations. Note that users should not forget to close file once data output is complete (dataset = None).

newrasterband(*dst_ds*, *rasterarray*, *band_num*, *NoDataVal*=`None`)

Accepts a GDAL dataset pointer, NumPy array, band number, [NoDataValue], and creates new band in specified file.

writerasterbands(*outfile*, *format*, *XSize*, *YSize*, *geotrans*, *epsg*, *NoDataVal*=`None`, **rasterarrays*)

Accepts Numpy arrays, outputfile string, format and geotranslation metadata and writes to file on disk.

writerasterband(*rasterarray*, *outfile*, *format*, *aXSize*, *aYSize*, *geotrans*, *epsg*, *NoDataVal*=`None`)

Legacy function for backwards compatability with older scripts. Use writerasterbands instead.

Accepts raster in Numpy 2D-array, outputfile string, format and geotranslation metadata and writes to file on disk.

wkt2epsg(*wkt*)

Accepts well known text of Projection/Coordinate Reference System and generates EPSG code.

band2txt(*band*, *outfile*)

Accepts NumPy array writes to specified text file on disk.

B.1.2 Variables

Name	Description
gdt2npy	**Value:** `{1: 'uint8', 2: 'uint16', 3: 'int16', 4: 'uint32', 5: 'in...`
npy2gdt	**Value:** `{'float32': 6, 'float64': 7, 'int16': 3, 'int32': 5, 'uin...`
gdt2struct	**Value:** `{1: 'B', 2: 'H', 3: 'h', 4: 'I', 5: 'i', 6: 'f', 7: 'd'}`
__package__	**Value:** `None`

B.1.3 Source code for RasterIO module

```
__version__ = "1.1.1"
#!/usr/bin/env python
# raster.py - module of raster handling functions using GDAL and NUMPY

import os, sys, struct
import numpy as np
import numpy.ma as ma
import osgeo.osr as osr
import osgeo.gdal as gdal
from osgeo.gdalconst import *

# Data type dictionaries - references from GDT's to other Python types.
# GDT -> Numpy
gdt2npy =       {
                        1:'uint8',
                        2:'uint16',
                        3:'int16',
                        4:'uint32',
                        5:'int32',
                        6:'float32',
                        7:'float64'

                }
# Numpy -> GDT
npy2gdt =       {
                        'uint8':1,
                        'uint16':2,
                        'int16':3,
                        'uint32':4,
                        'int32':5,
                        'float32':6,
                        'float64':7

                }

# GDT -> Struct
gdt2struct =    {
                        1:'B',
```

```
                        2:'H',
                        3:'h',
                        4:'I',
                        5:'i',
                        6:'f',
                        7:'d'
                }


# function to open GDAL raster dataset
def opengdalraster(filename):
        '''Accepts filename for GDAL compatible file and returns a GDAL
            pointer (dataset).'''
        dataset = gdal.Open(filename, GA_ReadOnly)
        if dataset != None:
                return dataset
        else:
                raise IOError


# function to read raster image metadata
def readrastermeta(dataset):
        '''Accepts GDAL raster dataset (pointer) and returns,
           gdal_driver, XSize, YSize, projection info(well known text)
           , geotranslation data.'''
                # get GDAL driver
        driver_short = dataset.GetDriver().ShortName
        driver_long = dataset.GetDriver().LongName
                # get projection
        proj_wkt = dataset.GetProjection()
                # get geotransforamtion parameters
        geotransform = dataset.GetGeoTransform()
                # geotransform[0] = top left x
                # geotransform[1] = w-e pixel resolution
                # geotransform[2] = rotation, 0 if image is "north up"
                # geotransform[3] = top left y
                # geotransform[4] = rotation, 0 if image is "north up"
                # geotransform[5] = n-s picel resolution
        XSize = dataset.RasterXSize
        YSize = dataset.RasterYSize
```

```
        return driver_short, XSize, YSize, proj_wkt, geotransform

# function to read a band from a dat# apply NoDataValue masking.aset
def readrasterband(dataset, aband, NoDataVal=None, masked=True):
        '''Accepts GDAL raster dataset (pointer) and band number,
           returns Numpy array.'''
        if dataset.RasterCount >= aband:
                # Get one band
                band = dataset.GetRasterBand(aband)
                # test for user specified input NoDataValue
                if NoDataVal is None:
                        # test for band specified NoDataValue
                        if band.GetNoDataValue() != None:
                                NoDataVal = band.GetNoDataValue()
                        #       print NoData
                        else:
                                # else set NoDataValue to be 9999.
                                NoDataVal = 9999
                # set NoDataVal for the band (not strictly needed, but
                   good practice if we call the band later)
                band.SetNoDataValue(NoDataVal)
                # create blank array (full of 0's) to hold extracted
                   data [note Y,X format], get data type from
                   dictionary
        # note that band is a GDAL pointer.
                datarray = np.zeros(( band.YSize,band.XSize ), gdt2npy[
                   band.DataType])
                        # create loop based on YAxis (i.e. num rows)
                for i in range(band.YSize):
                        # read lines of band    using GDAL ReadRaster
                           function (xoffset, yoffset, cols, rows,
                           buf_xsize, buf_ysize, datatype)
                        scanline = band.ReadRaster( 0, i, band.XSize,
                           1, band.XSize, 1, band.DataType)
                        # unpack from binary representation using
                           Python struct module
          # struct.unpack(datatype * nuumber cols, data)
          # conversion between GDAL-Python datatype names is done using
             RasterIO dictionary gdt2struct
```

```
                        tuple_of_vals = struct.unpack(gdt2struct[band.
                            DataType] * band.XSize, scanline)
                        # tuple_of_floats = struct.unpack('f' * band.
                            XSize, scanline)
                        # add tuple to image array line by line
                        datarray[i,:] = tuple_of_vals

                # check if masked=True
                if masked is True:
                        # check if data type is int or float using
                            dictionary for numeric test.
                        if npy2gdt[datarray.dtype.name] <= 5:
                                # data is integer use masked_equal
                                # apply NoDataValue masking.
                                dataraster = ma.masked_equal(datarray,
                                    NoDataVal, copy=False)
                                # apply invalid data masking
                                dataraster = ma.masked_invalid(
                                    dataraster, copy=False)
                                return dataraster
                        else:
                                # data is float use masked_values
                                dataraster = ma.masked_values(datarray,
                                     NoDataVal, copy=False)
                                # finaly apply mask for NaN values
                                dataraster = ma.masked_invalid(
                                    dataraster, copy=False)
                                # return array (raster)
                                return dataraster
                else:
                        # user wants numpy array, no masking.
                        return datarray
        else:
                raise TypeError

# function to create new (empty) raster file on disk.
def newgdalraster(outfile, format, XSize, YSize, geotrans, epsg,
    num_bands, gdal_dtype ):
        '''Accepts filename, format, XSize, YSize, geotransformation,
            epsg, number_of_bands, gdal_datatype and returns gdal
```

```
        pointer to new file.
                                                                    136
    This is a lower level function that allows users to control
        data output stream directly, use for specialist cases such
        as varying band data types or memory limited read-write
        situations.
    Note that users should not forget to close file once data       138
        output is complete (dataset = None).'''
    # get driver and driver properties
    driver = gdal.GetDriverByName( format )                         140
    metadata  = driver.GetMetadata()
    # check that specified driver has gdal create method and go     142
        create
    if metadata.has_key(gdal.DCAP_CREATE) and metadata[gdal.
        DCAP_CREATE] =='YES':
            # Create file                                           144
            dst_ds = driver.Create( outfile, XSize, YSize,
                num_bands, gdal_dtype )
            # define "srs" as a home for coordinate system          146
                parameters
            srs = osr.SpatialReference()
            # import the standard EPSG ProjCRS                      148
            srs.ImportFromEPSG( epsg )
            # apply the geotransformation parameters                150
            #print geotrans
            dst_ds.SetGeoTransform( geotrans )                      152
            # export these features to embedded well Known Text in
                the GeoTiff
            dst_ds.SetProjection( srs.ExportToWkt() )               154
            return dst_ds
    # catch error if no write method for format specified           156
    else:
            #print 'Error, GDAL %s driver does not support Create()  158
                method.' % outformat
            raise TypeError
                                                                    160
def newrasterband(dst_ds, rasterarray, band_num, NoDataVal=None):
    '''Accepts a GDAL dataset pointer, NumPy array, band number, [  162
        NoDataValue], and creates new band in specified file.'''
    # first check whether array is masked
```

```
        if ma.isMaskedArray(rasterarray) is True:                    164
                if NoDataVal is None:
                        if npy2gdt[rasterarray[0].dtype.name] == 1:   166
                                NoDataVal = 0
                        else:                                        168
                                NoDataVal = 9999
                dst_ds.GetRasterBand(band_num).SetNoDataValue(NoDataVal  170
                    )
                # create a numpy view on the masked array
                output = np.array(rasterarray, copy=False)           172
                # check if maskedarray has valid mask and apply to
                    numpy array using binary indexing.
                if rasterarray.mask is not ma.nomask:                174
                        output[rasterarray.mask] = NoDataVal
                # write out numpy array with masking                 176
                dst_ds.GetRasterBand(band_num).WriteArray ( output )
        else:                                                        178
        # input array is numpy already, write array to band in file
                dst_ds.GetRasterBand(band_num).WriteArray ( rasterarray  180
                    )

# create function to write GDAL rasters from NumPy arrays              182
def writerasterbands(filename, format, XSize, YSize, geotrans, epsg,
    NoDataVal=None, *rasterarrays ):
        ''' Accepts Numpy arrays, outputfile string, format and      184
            geotranslation metadata and writes to file on disk.'''
        # get number of bands
        num_bands = len(rasterarrays)                                186
        # create new raster using newgdal raster(output filename, GDAL
            drive, cols, rows, geotranslation_params, EPSG_code,
            num_bands, output_datatype (using rasterIO npy2gdt dict).
        dst_ds = newgdalraster(filename, format, XSize, YSize, geotrans  188
            , epsg, num_bands, npy2gdt[rasterarrays[0].dtype.name])
        # add raster data from raster NumPy arrays
        band_num = 1 # band counter                                  190
        for band in rasterarrays:
                newrasterband(dst_ds, band, band_num, NoDataVal)     192
                band_num += 1
        # close output and flush cache to disk                       194
        dst_ds = None
```

```
                                                                    196
# legacy function to write GeoTiff raster from NumPy n-dimensional
    array - use writerasterbands instead
def writerasterband(rasterarray, outfile, format, aXSize, aYSize,   198
    geotrans, epsg, NoDataVal=None):
        ''' Legacy function for backwards compatability with older
            scripts. Use writerasterbands instead.
                                                                    200
        Accepts raster in Numpy 2D-array, outputfile string, format and
             geotranslation metadata and writes to file on disk.'''
        writerasterbands(outfile, format, aXSize, aYSize, geotrans,  202
            epsg, NoDataVal, rasterarray)

# function to get Authority (e.g. EPSG) code from well known text     204
def wkt2epsg(wkt):
        '''Accepts well known text of Projection/Coordinate Reference  206
            System and generates EPSG code.'''
        if wkt is not None:
                if wkt == '':                                        208
                        return 0
                else:                                                210
                        srs = osr.SpatialReference(wkt)
                        if (srs.IsProjected()):                      212
                                return int(srs.GetAuthorityCode("PROJCS
                                    "))
                        elif (srs.IsLocal()):                        214
                                return 0
                        else:                                        216
                                return int(srs.GetAuthorityCode("GEOGCS
                                    "))
        else:                                                        218
                raise TypeError
def band2txt(band, outfile):                                         220
        '''Accepts NumPy array writes to specified text file on disk.
            '''
        if ma.isMaskedArray(band) is True:                           222
                outraster = ma.compressed(band)
        else:                                                        224
                outraster = band
        np.savetxt(outfile, outraster, fmt='%f')                     226
```

B.2 Documentation for the PyAVHRR module

PyAVHRR.py - Library of procedural processing functions for Advanced Very High Resolution Radiometer (AVHRR) data in Numpy array form.

Using the documentation

Documentation for module functions is provided as Python docstrings, accessible from an interactive Python terminal. Within docstrings examples from an iterative Python console are identified using '>>>'. Further information is given to developers within the source code using '#' comment strings. To view this text and a list of available functions call the Python in-built help command, specifying module name.

```
>>> import PyAVHRR as avhrr
>>> help(avhrr)
>>> ...this text...
```

For help on a specific function call the Python in-built help command, specifying module.function.

```
>>> import PyAVHRR as avhrr
>>> help(avhrr.ndvi)

>>> ndvi(b1, b2)
>>> Accepts two Numpy AVHRR bands 1 & 2,
>>> returns Normalised Differenced Vegetation Index.
```

How to access functions

To access functions, import the module to Python and call the desired function, assigning the output to a named variable. Note that the primary input data-type (default) for all functions is either a Numpy array or a Numpy masked array. Within this module the term "raster" is used to signify a Numpy/Numpy masked array of raster values. Use the rasterIO module to convert Numpy arrays to/from Geospatial raster data-formats.

```
>>> import PyAVHRR as avhrr
>>> band1, band2, band3, band4, band5 = avhrr.readbands(
>>>                                         gdal_file_pointer)
```

Note - to avoid programme conflicts it is best to not have common names for variables and functions. i.e. >>> my_ndvi_raster = avhrr.ndvi(b1, b2). Not: ndvi = avhrr.ndvi(b1, b2) In the docstrings variables are often prefixed with 'r' for raster (e.g. rsnow).

Dependencies

Python 2.6 or greater Numerical python (Numpy) 1.2.1 or greater

License & Authors

Copyright: Tom Holderness Development team: Tom Holderness, Andrew Hardy, Nathan Forsythe Released under GPL v2 (see license.txt).

Version: 1.3

B.2.1 Functions

readbands(*fpointer*)

Function to read an AVHRR scene, accepts GDAL pointer and returns one raster array per band (bands 1-5).

This function allows user to read all five bands of an AVHRR scene without repeated calls to rasterIO within their programme. User rasterIO.opengdalraster('scenename.ext') to get a GDAL file pointer for an AVHRR scene

```
>>> import rasterIO, avhrr
>>> scenepointer = rasterIO.opengdalraster(
>>>                                     'scene.tif')
>>> b1, b2, b3, b4, b5 = avhrr.readbands(
>>>                                 gdal_file_pointer)
```

ndvi(*b1*, *b2*)

Accepts AVHRR bands 1 & 2, returns Normalised Differenced Vegetation Index as new Numpy array.

Input may be albedo/at-sensor reflectance/at-surface reflectance. If either input band has been masked (e.g. cloud coverage), NDVI output will follow same mask. If both input bands have different masks (e.g. cloud coverage), NDVI output will have a mask based on a union join both input masks.

```
>>> fndvi = avhrr.ndvi(b1, b2)
```

cloudmask(*r1*, *rcloud*)

Accepts a boolean raster of cloud coverage (where cloud pixel = 1) and masks another given raster

Note that both rasters must cover the same location and have the same spatial resolution and dimensions.

lst_DLR(*b4*, *b5*, *ndvi*)

Accepts a NDVI raster and AVHRR bands 4 and 5, derives a new raster of estimated land surface temperature (Kelvin).

This function calculates estimated land surface temperature using a split window method to correct atmospheric attenuation of thermal bands and NDVI to apply a correct for surface emissivity. Returns a new single raster of Estimated surface temperature in Kelvin.

Developed by Deutsches Zentrum fur Luft- und Raumfahrt (DLR) - German Aerospace Center Visit: http://eoweb.dlr.de/short_guide/D-LST.html for more details.

```
>>> rlst = avhrr.lst.DLR(b4, b5, fndvi)
```

rasterUHII(*raster*, *ruralXpix*, *ruralYpix*)

Accepts Numpy raster and returns UHII representation (urban_px - rural_px)

Function is named rasterUHII to differentiate from more traditional air-temperature derived UHII values found in literature.

```
>>> surf_uhii = avhrr.rasterUHII(rlst, 850, 900)
```

B.2.2 Source code for AVHRR module

```
"""
avhrr.py - Library of procedural processing functions for
Advanced Very High Resolution Radiometer (AVHRR) data in Numpy.Ma
    matrix form.
"""
#!/usr/bin/env python

__version__ = "1.3"

# Import commands
import sys, os
import numpy as np
import numpy.ma as ma
import datetime as datetime
from datetime import time as time
import rasterIO

# Read all five bands in an AVHRR scene to Numerical Python arrays.
def readbands(fpointer):
        '''Function to read an AVHRR scene, accepts GDAL pointer and
            returns one raster array per band (bands 1-5).

        This function allows user to read all five bands of an AVHRR
            scene without repeated calls to rasterIO
        within their programme.
        User rasterIO.opengdalraster('scenename.ext') to get a GDAL
            file pointer for an AVHRR scene

        >>> import rasterIO, avhrr
        >>> scenepointer = rasterIO.opengdalraster('scene.tif')
        >>> band1, band2, band3, band4, band5 = avhrr.readbands(
            gdal_file_pointer)

        '''
        b1 = rasterIO.readrasterband(fpointer, 1)
        b2 = rasterIO.readrasterband(fpointer, 2)
        b3 = rasterIO.readrasterband(fpointer, 3)
        b4 = rasterIO.readrasterband(fpointer, 4)
```

```
        b5 = rasterIO.readrasterband(fpointer, 5)
        return b1, b2, b3, b4, b5

# Function to calculate NDVI from AVHRR bands 1 and 2 (requires np array of albedo/at sensor reflectance/at surface reflectance)
def ndvi(b1, b2):
        '''Accepts AVHRR bands 1 & 2, returns Normalised Differenced Vegetation Index as new Numpy array.

        Input may be albedo/at-sensor reflectance/at-surface reflectance.
        If either input band has been masked (e.g. cloud coverage), NDVI output will follow same mask.
        If both input bands have different masks (e.g. cloud coverage), NDVI output will have a mask based on a union join both input masks.

        >>> fndvi = avhrr.ndvi(b1, b2)
        '''
        step1=(b2-b1)
        step2=(b2+b1)
        ndvi_raster=step1/step2
        return ndvi_raster

# Function apply cloud mask to input raster.
def cloudmask(r1, rcloud):
        '''Accepts a boolean raster of cloud coverage (where cloud pixel = 1) and masks another given raster

        Note that both rasters must cover the same location and have the same spatial resolution and dimensions.

        '''
        # Create mask from cloud raster (1 = cloud)
        clouds= ma.masked_values(rcloud, 1)
        # Extract mask from clouds raster
        cloudmask = ma.getmask(clouds)
        # Apply new mask to input raster (Union to existing raster).
        new_r1 = ma.masked_array(r1, mask=cloudmask)
        return new_r1
```

```
                                                                    67
# Function to calculate estimated land surface temperature from AVHRR
    ch4/ch5 (see http://eoweb.dlr.de/short_guide/D-LST.html)
def lst_DLR (b4, b5, ndvi):                                         69
        '''Accepts a NDVI raster and AVHRR bands 4 and 5, derives a new
            raster of estimated land surface temperature (Kelvin).
                                                                    71
        This function calculates estimated land surface temperature
            using a split window method to correct
        atmospheric attenuation of thermal bands and NDVI to apply a    73
            correct for surface emissivity.
        Returns a new single raster of Estimated surface temperature in
            Kelvin.
                                                                    75
        Developed by Deutsches Zentrum fur Luft- und Raumfahrt (DLR) -
            German Aerospace Center
        Visit: http://eoweb.dlr.de/short_guide/D-LST.html for more      77
            details.

        >>> rlst = avhrr.lst.DLR(b4, b5, fndvi)                         79
        '''
        # first scale any NDVI inputs                                   81
        ndvi255 = (ndvi+1)*127
        # second, calculate coefficients                                83
        e4 = 1.0094 + 0.047*(np.log(ndvi255))
        e5 = e4 + 0.01                                                  85
        e = (e4 + e5)/2
        de = e4 - e5                                                    87
        lst_raster = 1.274+(b4+b5)/2*(1+0.15616*((1-e)/e)-0.482*de/(np.
            power(e,2)))+(b4-b5)/2*(6.26+3.989*((1-e)/e)+38.33*de/(np.
            power(e,2)))
        return lst_raster                                               89

# Function to calculate surface UHII based on values from a rural pixel  91
    within scene
def rasterUHII(raster, ruralXpix, ruralYpix):
        '''Accepts Numpy raster and returns UHII representation (       93
            urban_px - rural_px)

        Function is named rasterUHII to differentiate from more         95
```

```
   traditional air-temperature
derived UHII values found in literature.
                                                                    97
>>> surf_uhii = avhrr.rasterUHII(rlst, 850, 900)
'''                                                                 99

return raster - (getpixelval(raster, ruralXpix, ruralYpix))       101
```

B.3 Source code for the Raster Processing Suite

```
"""
*********************************************************************
Name:   Raster Processing Suite
Description:    Perform raster math calculations in QGIS.
Date:   05/10/2010
copyright:  (C) 2010 Tom Holderness & Newcastle University
contact:    http://www.students.ncl.ac.uk/tom.holderness
license:     Relseased under Simplified BSD license (see LICENSE.txt)
 *********************************************************************
"""
# Note: change log moved to seperate file for brevity

# This file contains the core functionality for the QGIS RPS plugin.
# Other files required for plugin:
#  - __init__.py - standard QGIS init file
#  - PyRasterPlugin.py = handle loading of plugin in QQGIS
#  - rasterIO.py - perform raster input/output
#  - rasterProcessor_ui.py - Python representation of Qt GUI XML
#  - LICENSE.TXT contains software license

# Import the PyQt libraries
from PyQt4 import QtCore, QtGui
# Import standard libraries
import sys, os, string
from os.path import isfile
# Import QGIS and QGIS core
from qgis.core import *
import qgis
# Initialize Qt resources from file resources.py
import resources
# Import the dialog
from rasterProcessor_ui import Ui_Form
# rasterIO and associates
import rasterIO
import numpy.ma as ma
import numpy as np
from datetime import datetime
# Matplotlib for plotting histograms (experimental)
```

```
import matplotlib
matplotlib.use('Qt4Agg')
import matplotlib.pyplot as plt

# Setup QGIS plugin.
import __init__ as initfile
version = initfile.version
rasterIO_version = rasterIO.__version__

# Classes for redicreting stdout and stderr to the plugin window
class StdOutLog:

    def __init__(self, edit, out):
        """
        http://www.riverbankcomputing.com/pipermail/pyqt/2009-February
            /022025.html
        """
        self.edit = edit
        self.out = out

    def write(self, m):
            self.edit.setTextColor(QtCore.Qt.black)
        self.edit.insertPlainText( m )
        self.edit.moveCursor(QtGui.QTextCursor.End)

class StdErrLog:

    def __init__(self, edit, out):
        """
        http://www.riverbankcomputing.com/pipermail/pyqt/2009-February
            /022025.html
        """
        self.edit = edit
        self.out = out

    def write(self, m):
            self.edit.setTextColor(QtCore.Qt.red)
        self.edit.insertPlainText( m )
        self.edit.moveCursor(QtGui.QTextCursor.End)
```

```
# Class containing the raster processing suite, inheriting from Qt QDialog
class RasterProcessingSuite(QtGui.QDialog):
    def __init__(self):
      # Set up Qt stuff for plugin GUI
        QtGui.QDialog.__init__(self)
        self.ui = Ui_Form ()
        self.ui.setupUi(self)
        # Capture stdout and stderr and send to plugin console
        sys.stdout = StdOutLog( self.ui.textInformation, sys.stdout)
        sys.stderr = StdErrLog( self.ui.textInformation, sys.stderr)
      # Initialise console with date and time for userr
        sys.stdout.write(str(datetime.now().strftime("%d-%m-%Y %H:%M\n"
            )))
        # Connect the signals and slots of the widget GUI elements
      # List of layers loaded in QGIS
        QtCore.QObject.connect(self.ui.listWidget_Layers,QtCore.SIGNAL(
            "itemClicked(QListWidgetItem*)"),self.get_band_list)
        QtCore.QObject.connect(self.ui.listWidget_Layers,QtCore.SIGNAL(
            "itemChanged(QListWidgetItem*)"),self.get_band_list)
      # Load band list and button
        QtCore.QObject.connect(self.ui.btnLoad,QtCore.SIGNAL("pressed()
            "),self.load_band_status)
        QtCore.QObject.connect(self.ui.btnLoad,QtCore.SIGNAL("released
            ()"),self.load_band)
      # Run, clear, save buttons
        QtCore.QObject.connect(self.ui.btnRun,QtCore.SIGNAL("pressed()"
            ),self.run_status)
        QtCore.QObject.connect(self.ui.btnRun,QtCore.SIGNAL("released()
            "),self.run)
        QtCore.QObject.connect(self.ui.btnClear,QtCore.SIGNAL("pressed
            ()"),self.clear_EqEdit)
        QtCore.QObject.connect(self.ui.btnSave,QtCore.SIGNAL("clicked()
            "),self.save_file_dialog)
      # Output checkbox
        QtCore.QObject.connect(self.ui.checkBoxGenerateOutput,QtCore.
            SIGNAL("toggled(bool)"),self.disable_output)
        # Calculator button operators
        QtCore.QObject.connect(self.ui.btnAddition,QtCore.SIGNAL("
```

```
    clicked()"),self.insertAdd)
QtCore.QObject.connect(self.ui.btnSubtract,QtCore.SIGNAL("
    clicked()"),self.insertMinus)
QtCore.QObject.connect(self.ui.btnDivide,QtCore.SIGNAL("clicked
    ()"),self.insertDivide)
QtCore.QObject.connect(self.ui.btnMultiply,QtCore.SIGNAL("
    clicked()"),self.insertMult)
QtCore.QObject.connect(self.ui.btnSqRoot,QtCore.SIGNAL("clicked
    ()"),self.insertRoot)
QtCore.QObject.connect(self.ui.btnSquared,QtCore.SIGNAL("
    clicked()"),self.insertPower)
QtCore.QObject.connect(self.ui.btnRBracket,QtCore.SIGNAL("
    clicked()"),self.insertRbracket)
QtCore.QObject.connect(self.ui.btnLBracket,QtCore.SIGNAL("
    clicked()"),self.insertLbracket)
QtCore.QObject.connect(self.ui.btnMean,QtCore.SIGNAL("clicked()
    "),self.insertMean)
QtCore.QObject.connect(self.ui.btnStDev,QtCore.SIGNAL("clicked
    ()"),self.insertStDev)
QtCore.QObject.connect(self.ui.btn0,QtCore.SIGNAL("clicked()"),
    self.insertZero)
QtCore.QObject.connect(self.ui.btn1,QtCore.SIGNAL("clicked()"),
    self.insertOne)
QtCore.QObject.connect(self.ui.btn2,QtCore.SIGNAL("clicked()"),
    self.insertTwo)
QtCore.QObject.connect(self.ui.btn3,QtCore.SIGNAL("clicked()"),
    self.insertThree)
QtCore.QObject.connect(self.ui.btn4,QtCore.SIGNAL("clicked()"),
    self.insertFour)
QtCore.QObject.connect(self.ui.btn5,QtCore.SIGNAL("clicked()"),
    self.insertFive)
QtCore.QObject.connect(self.ui.btn6,QtCore.SIGNAL("clicked()"),
    self.insertSix)
QtCore.QObject.connect(self.ui.btn7,QtCore.SIGNAL("clicked()"),
    self.insertSeven)
QtCore.QObject.connect(self.ui.btn8,QtCore.SIGNAL("clicked()"),
    self.insertEight)
QtCore.QObject.connect(self.ui.btn9,QtCore.SIGNAL("clicked()"),
    self.insertNine)
QtCore.QObject.connect(self.ui.btnPoint,QtCore.SIGNAL("clicked
```

```
        ()"),self.insertPoint)
    # Tab_2 buttons (Python scripts)                                    126
    QtCore.QObject.connect(self.ui.btnClearScript,QtCore.SIGNAL("
        clicked()"),self.clear_Pyout)
    QtCore.QObject.connect(self.ui.btnRunScript,QtCore.SIGNAL("       128
        pressed()"),self.run_status)
    QtCore.QObject.connect(self.ui.btnRunScript,QtCore.SIGNAL("
        released()"),self.run_Pyout)
    QtCore.QObject.connect(self.ui.btnSaveScript,QtCore.SIGNAL("      130
        clicked()"),self.save_Pyout)
    QtCore.QObject.connect(self.ui.btnOpenScript,QtCore.SIGNAL("
        clicked()"),self.open_Pyout)
    QtCore.QObject.connect(self.ui.btnOpenTemplate,QtCore.SIGNAL("    132
        clicked()"),self.open_template)
    # Tab_3 buttons (information)
    QtCore.QObject.connect(self.ui.btnViewLicense,QtCore.SIGNAL("     134
        clicked()"),self.print_license)
    # Get the rasters that have been loaded into QGIS
    self.add_band()                                                   136
    # Tab_2 setup script window
    self.ui.textPyout.insertPlainText('#!/usr/bin/env python\n')      138
    self.ui.textPyout.insertPlainText('import rasterIO\n')
    self.ui.textPyout.insertPlainText('import numpy.ma as ma\n\n')    140
# Add band function to add a available raster bands loaded in QGIS
   to list of available layers in the plugin.
def add_band(self):                                                   142
    self.layermap = QgsMapLayerRegistry.instance().mapLayers()
  # Loop over QGIS layermap to add rasters                            144
    for (name, layer) in self.layermap.iteritems():
        if type(layer).__name__ == "QgsRasterLayer":                  146
            raster =  layer.source()
            try:                                                      148
            # Try and create a pointer to each raster to get number
                of bands
                raster_str = str(raster)                              150
                rasterIO.opengdalraster(raster_str)
                self.ui.listWidget_Layers.addItem(raster_str)         152
                self.ui.listWidget_Layers.setCurrentRow(0)
                self.get_band_list()                                  154
        # Catch IO errors on GDAL pointer opening
```

```
            except IOError:
                sys.stderr.write('IOError from file: ')
                sys.stderr.write(raster_str)
                sys.stderr.write('\n')
# write - utility method to send text data to the plugin console
 def write(self,astring):
     self.ui.textInformation.append(astring)
# exit method
 def exit(self):
     quit()
# load band status - utility function to inform the user of band
   loading
 def load_band_status(self):
     sys.stdout.write('Reading raster data...\n')
 # Maths methods for calculator functions, add the text to each
    function to the equation editor
# divide
 def insertStDev(self):
     self.ui.textEqEdit.insertPlainText(" ma.std( )")
     self.ui.textEqEdit.moveCursor(QtGui.QTextCursor.
         PreviousCharacter)
# brackets
 def insertRbracket(self):
     self.ui.textEqEdit.insertPlainText(") ")
 def insertLbracket(self):
     self.ui.textEqEdit.insertPlainText("( ")
# square root
 def insertRoot(self):
     self.ui.textEqEdit.insertPlainText("ma.sqrt( )")
     self.ui.textEqEdit.moveCursor(QtGui.QTextCursor.
         PreviousCharacter)
# square
 def insertPower(self):
     self.ui.textEqEdit.insertPlainText("ma.power( ,2)")
     self.ui.textEqEdit.moveCursor(QtGui.QTextCursor.
         PreviousCharacter)
     self.ui.textEqEdit.moveCursor(QtGui.QTextCursor.
         PreviousCharacter)
     self.ui.textEqEdit.moveCursor(QtGui.QTextCursor.
         PreviousCharacter)
```

```
# mean average
 def insertMean(self):                                                  190
     self.ui.textEqEdit.insertPlainText(" ma.mean( )")
     self.ui.textEqEdit.moveCursor(QtGui.QTextCursor.                   192
         PreviousCharacter)
# Add/Minus/Divide/Multiply
 def insertAdd(self):                                                   194
     self.ui.textEqEdit.insertPlainText("+ ")
 def insertMinus(self):                                                 196
     self.ui.textEqEdit.insertPlainText("- ")
 def insertDivide(self):                                                198
     self.ui.textEqEdit.insertPlainText("/ ")
 def insertMult(self):                                                  200
     self.ui.textEqEdit.insertPlainText("* ")
# Numbers and decimal point                                            202
 def insertZero(self):
     self.ui.textEqEdit.insertPlainText("0")                            204
 def insertOne(self):
     self.ui.textEqEdit.insertPlainText("1")                            206
 def insertTwo(self):
     self.ui.textEqEdit.insertPlainText("2")                            208
 def insertThree(self):
     self.ui.textEqEdit.insertPlainText("3")                            210
 def insertFour(self):
     self.ui.textEqEdit.insertPlainText("4")                            212
 def insertFive(self):
     self.ui.textEqEdit.insertPlainText("5")                            214
 def insertSix(self):
     self.ui.textEqEdit.insertPlainText("6")                            216
 def insertSeven(self):
     self.ui.textEqEdit.insertPlainText("7")                            218
 def insertEight(self):
     self.ui.textEqEdit.insertPlainText("8")                            220
 def insertNine(self):
     self.ui.textEqEdit.insertPlainText("9")                            222
 def insertPoint(self):
     self.ui.textEqEdit.insertPlainText(".")                            224
# Method to set GUI options when output file is disabled
 def disable_output(self):                                              226
     if self.ui.btnSave.isEnabled() == True:
```

```
            self.ui.btnSave.setEnabled(False)                                  228
            self.ui.lineOutfile.setEnabled(False)
            self.ui.comboFormats.setEnabled(False)                             230
            self.ui.checkBoxQGIS.setEnabled(False)
            self.ui.labelSaveNewRaster.setEnabled(False)                       232
        else:
            self.ui.btnSave.setEnabled(True)                                   234
            self.ui.lineOutfile.setEnabled(True)
            self.ui.comboFormats.setEnabled(True)                              236
            self.ui.checkBoxQGIS.setEnabled(True)
            self.ui.labelSaveNewRaster.setEnabled(True)                        238
    # get_band_list - method to get a list of bands available in the
       raster
    def get_band_list(self):                                                   240
        if (self.ui.listWidget_Layers.count() > 0):
            fname = self.ui.listWidget_Layers.currentItem().text()             242
            fname_Str = str(fname)
            inraster = rasterIO.opengdalraster(fname_Str)                      244
            numbands = inraster.RasterCount
            self.ui.comboBands.clear()                                         246
            self.ui.comboBands.addItem("Band #")
            for i in range(1,numbands +1):                                     248
                self.ui.comboBands.addItem(str(i))
            self.ui.comboBands.setCurrentIndex(1)                              250
    # Load the band into memory as NumPy array using RasterIO
    def load_band(self):                                                       252
        file_count = self.ui.listWidget_Layers.count()
        if file_count < 1:                                                     254
         # Catch for broken layer
            sys.stderr.write('Error: No input files, open a raster file        256
                first!\n')
        else:
         # Catch for user to select band number                                258
            if self.ui.comboBands.currentText() == 'Band #':
                sys.stderr.write('Select which band number to load.\n')        260
            else:
            # Proceed to load file and perform file name formatting for        262
                equation editor (e.g. avhrr.tif band 1 would be '
               avhrr_1')
                band_num = int(self.ui.comboBands.currentText())
```

```
        fname = self.ui.listWidget_Layers.currentItem().text()
        fname_Str = str(fname)
        basename = os.path.basename(fname_Str)
        basename = os.path.splitext(basename)
        basename = basename[0]
        cleaname_a = string.replace(basename, '.', '_')
        cleaname = string.replace(cleaname_a, '-', '_')
        newname = cleaname+'_'+str(band_num)
    # Open GDAL pointer with RasterIO
        rasterpointer = rasterIO.opengdalraster(fname_Str)
    # Define global space variables for this raster properties
       so can be accessed by all GUI methods
        global driver, XSize, YSize, proj, geotrans
        global bandname
    # Assign bandname to the name representation in the
       equation editor
        bandname = newname
        # Check memory to see if band already loaded (new v
           0.9.4) stops loading same band more than once
        if bandname not in globals():
       # If band doesn't exist load as global with predifined
          name
            globals()[bandname] = rasterIO.readrasterband(
               rasterpointer, band_num)
       # Get geospatial meta-data
            driver, XSize, YSize, proj, geotrans = rasterIO.
               readrastermeta(rasterpointer)
    # Send appropriate information to the GUI after load
        self.ui.textEqEdit.insertPlainText(bandname+" ")
        sys.stdout.write("Loaded: ")
        sys.stdout.write(str(fname_Str))
        sys.stdout.write(", band: ")
        sys.stdout.write(str(band_num))
        sys.stdout.write("\n")
    # Populate the script editor with the commands to open the
       specified file
        self.ui.textPyout.insertPlainText('# open a file
           pointer\n')
        self.ui.textPyout.insertPlainText('rasterpointer =
           rasterIO.opengdalraster("%s")\n' %(fname_Str))
```

```
                self.ui.textPyout.insertPlainText('# read a raster band
                    \n')
                self.ui.textPyout.insertPlainText('%s = rasterIO.
                    readrasterband(rasterpointer, %i)\n' %(bandname,
                    band_num))
                self.ui.textPyout.insertPlainText('# get file metadata:
                     format, X, Y, projection, geo-parameters\n')
                self.ui.textPyout.insertPlainText('driver, XSize, YSize
                    , proj, geotrans = rasterIO.readrastermeta(
                    rasterpointer)\n\n')
# run statis - utility method to inform user that process is
    running
def run_status(self):
    sys.stdout.write('Processing...\n')
# Run - Execute the equation in the equation editor
def run(self):
    # Get the inputs
    outname = str(self.ui.lineOutfile.text())
    eqstring = str(self.ui.textEqEdit.toPlainText())
    # Basic user validation of equation
    if (len(eqstring) < 1):
        sys.stderr.write('Error: No equation to process.\n')
    elif(self.ui.listWidget_Layers.count() < 1):
        sys.stderr.write('Error: No input files.\n')
    # Process to new file
    else:
        try:
            # Test if output box is checked
            if self.ui.checkBoxGenerateOutput.isChecked() == False:
                if (len(outname) < 1):
                    sys.stderr.write('Error: No output filename
                        specified.\n')
                else:
                  # Use eval to translate equation editor string into
                     Python command
                    newband = eval(eqstring)
                    newband = ma.masked_values(newband, 9999.0)
                    epsg = rasterIO.wkt2epsg(proj)
                    # Setup python dictionary of rgdal formats and
                        drivers
```

```
            formats = {'GeoTiff (.tif)':'.tif','Erdas
                Imagine (.img)':'.img'}
            drivers = {'GeoTiff (.tif)':'GTiff','Erdas
                Imagine (.img)':'HFA'}
            out_ext = formats[str(self.ui.comboFormats.
                currentText())]
            driver = drivers[str(self.ui.comboFormats.
                currentText())]
            outfile = outname + out_ext
        # Write the result of the calculation to a new raster
            file using RasterIO
            rasterIO.writerasterband(newband, outfile,
                driver, XSize, YSize, geotrans, epsg)
        # Update the user on the situation
            sys.stdout.write('Process complete, created
                newfile ')
            sys.stdout.write(str(outfile))
            sys.stdout.write('\n')
            if self.ui.checkBoxQGIS.isEnabled() == True:
           # Add the new file to QGIS layers and canvas if
               required
                qgis.utils.iface.addRasterLayer(outfile)
        # Populate the script editor with output commands for
            the the calculation
            self.ui.textPyout.insertPlainText('# create a
                new matrix from equation\n')
            self.ui.textPyout.insertPlainText('newband = %s
                \n' %(eqstring))
            self.ui.textPyout.insertPlainText('# get the
                epsg code from the projection\n')
            self.ui.textPyout.insertPlainText('epsg =
                rasterIO.wkt2epsg(proj)\n')
            self.ui.textPyout.insertPlainText('# set the
                gdal driver / output file type\n')
            self.ui.textPyout.insertPlainText('driver = "%s
                "\n' %(driver))
            self.ui.textPyout.insertPlainText('# specify
                the new output file\n')
            self.ui.textPyout.insertPlainText('outfile = "%
                s"\n' %(outfile))
```

```
                    self.ui.textPyout.insertPlainText('# write the
                        new matrix to the new file\n')
                    self.ui.textPyout.insertPlainText('rasterIO.
                        writerasterband(newband, outfile, driver,
                        XSize, YSize, geotrans, epsg)\n\n')
                    self.ui.textPyout.insertPlainText('# add the
                        new file to qgis\n')
                    self.ui.textPyout.insertPlainText('qgis.utils.
                        iface.addRasterLayer(outfile)\n\n')
                else:
                # Do no create ouput, perform calculation in memory and
                    return output to plugin console
                    outputstring = (str(eval(str(self.ui.textEqEdit.
                        toPlainText())))) +'\n'
                    self.ui.textInformation.setTextColor(QtGui.QColor
                        (0,0,255))
                    self.ui.textInformation.insertPlainText(
                        outputstring)
                    self.ui.textInformation.moveCursor(QtGui.
                        QTextCursor.End)
                    self.ui.textPyout.insertPlainText('# run without
                        output file\n')
                # Print result of equation
                    self.ui.textPyout.insertPlainText('print %s\n\n' %(
                        eqstring))
            # Error catches for user raster calculaltions
            except ValueError:
                sys.stderr.write('Error: Could not perform calculation.
                    Are input rasters same shape and size? Is the
                    output a matrix?\n')
            except TypeError:
                sys.stderr.write('Error: Could not perform calculation.
                    Are input rasters loaded?\n')
            except SyntaxError:
                sys.stderr.write('Error: Could not perform calculation.
                    Is the equation correct?\n')
            except AttributeError:
                sys.stderr.write('Error: Could not perform calculation.
                    Is the output raster correct?\n')
    # Python script functions for Tab 2.
```

```
 def run_Pyout(self):
     try:                                                          372
      # Try to run the script in the script editor window
         commandstring = str(self.ui.textPyout.toPlainText())      374
         exec(commandstring)
     except:                                                       376
         sys.stderr.write('Error: There was an error in the script.\
            n')
# Utility function to open Qt save file dialog                    378
 def save_file_dialog(self):
     fd = QtGui.QFileDialog.getSaveFileName(self)                  380
     self.ui.lineOutfile.insert(fd)
# Functions to clear the equation editor and script window        382
 def clear_EqEdit(self):
     self.ui.textEqEdit.clear()                                    384
 def clear_Pyout(self):
     self.ui.textPyout.clear()                                     386
     self.ui.textPyout.insertPlainText('#!/usr/bin/env python\n')
     self.ui.textPyout.insertPlainText('import rasterIO\n')        388
     self.ui.textPyout.insertPlainText('import numpy.ma as ma\n\n')
#save_Pyout - allow the user to save the script in the script window 390
    to a Python file
 def save_Pyout(self):
     try:                                                          392
         fd = QtGui.QFileDialog.getSaveFileName(self, "Save script",
              "", "Python files: *.py")
         if fd != '':                                              394
              fd = fd+'.py'
              outfile = open(fd,'w')                               396
              print >> outfile, str(self.ui.textPyout.toPlainText())
              sys.stdout.write('Saved script to file\n')            398
     except IOError:
         sys.stderr.write('Error: There was an error saving the    400
            python script\n')
 # Allow the user to load a script into the script editor window
 def open_Pyout(self):                                             402
     try:
         fd = QtGui.QFileDialog.getOpenFileName(self, "Open script", 404
              "", "Python files: *.py ;; Text files: *.txt ;; All
            files: *.*")
```

```
            if fd != '':
                infile = open(fd,'r')
                script = infile.read()
                self.ui.textPyout.clear()
                self.ui.textPyout.insertPlainText(script)
        except IOError:
            sys.stderr.write('Error: There was an error opening the
               python script\n')
    # Open the script template from the plugin installation directory
    def open_template(self):
        try:
            cdir = os.getcwd()
            template_dir = os.environ['HOME']+'/.qgis/python/plugins/
               raster_processing_suite/templates/'
            os.chdir(template_dir)
            fd = QtGui.QFileDialog.getOpenFileName(self, "Open template
                script", "", "Python files: *.py ;; Text files: *.txt
               ;; All files *.*")
            if fd != '':
                infile = open(fd,'r')
                script = infile.read()
                self.ui.textPyout.clear()
                self.ui.textPyout.insertPlainText(script)
            os.chdir(cdir)
        except IOError:
            sys.stderr.write('Error: There was an error opening the
               python script\n')
    # Print the license in the console window
    def print_license(self):
        try:
            # read license file.
            license = open(os.environ['HOME']+'/.qgis/python/plugins/
               raster_processing_suite/LICENSE.TXT')
            text = license.read()
            # set license text colour to dark blue.
            self.ui.textInformation.setTextColor(QtCore.Qt.darkBlue)
            # write license to text file.
            self.ui.textInformation.insertPlainText(text)
        except IOError:
            sys.stdout.write("Error: Can't open LICENSE.TXT'.")
```

Appendix C

A parser for MIDAS stations data

This Bash script parses raw MIDAS station data and converts it into a series of SQL insert statements to insert into an empty database table. Station location geometry is encoded using the PostGIS GeomFromText function.

C.1 Parser source code

```
# MIDAS station parser

# Convert raw MIDAS CSV file to SQL insert and PostGIS Extended Well Known Text format
    for input into a PostGIS database

echo "BEGIN;"

gawk -F , 'a="'"'"'" {print "INSERT INTO \"STATIONS_GLA\" (PKEY, src_id, met_domain,
    id_type, id, src_cap_bgn, src_cap_end, src_bgn, src_end, stn_name, pcode,easting,
    northing,elevation,geom)\n" "VALUES" "("$1$20","$1","a$5a","a$3a","a$4a","a$6a","
    a$7a","a$18a","a$19a","a$2a","a$9a","a$13a","a$14a","a$15"," "ST_GeomFromEWKT("a"
    SRID=27700;POINT(" $13" "$14" " $15")"a"));"}'
echo "COMMIT;"
```

C.2 Parser execution

```
bash$: midas_station_parser.sh < midas_stations.csv > midas_stations.sql
```

Appendix D

A parser for MIDAS air temperature measurements

This Bash script parses raw MIDAS hourly measurement files (one file per year) and coverts them into series of SQL insert statements to insert into an empty database table.

D.1 Parser source code

```
#!/bin/bash

#midas_air_parser.sh
#Extract air temperature from BADC MIDAS WXHRLY
#a.) construct loop for input file names
#b.) make output dir in /tmp (needs to be temp for Postgres user access)
#b.) gawk for each file
#       i.)     remove trailing comma from air_temp
#       ii.)    insert comma between date & time
#       iii.)   output to outfile

inf=$1 #infile is first input file
outf=$2 #one large file for all output years
#outdir="/tmp/airtemp_WXHRLY_1985-2008/"
#mkdir $outdir
for (( i = 1985; i <= 2008; i++ )); do
        inf=midas_wxhrly_"$i"01-"$i"12.txt
        gawk  '{gsub(",","",$37) ; print ($1 "," $2 $3 $4 $6 $7 $37)}' $inf >> $outf
        echo $inf
```

```
done
echo Output: $outf
exit 0
```

D.2 Parser execution

```
bash$: midas_air_parser.sh < midas_air_data.csv > midas_air_data.sql
```

Appendix E

SQL queries for removing duplicate observations

This appendix contains the SQL statements used to filter null and erroneous air temperature measurements from the MIDAS terrestrial hourly air temperature data after it was loaded into the database.

E.1 SQL Queries

```
 '1.) Select non-null air temperature values to new table'
SELECT * INTO airtemp_hourly_uk_19852008_no_null
FROM airtemp_hourly_uk_19852008_clean_input
WHERE air_temp IS NOT NULL

'2.) Select records with version number 1 from non-null air
    temperature values to new table'
SELECT * INTO airtemp_hourly_uk_19852008_no_version0
FROM airtemp_hourly_uk_19852008_no_null
WHERE version_num = 1;

'3.) Select records without id_type ICAO to new table'
```

```
SELECT * INTO airtemp_hourly_uk_19852008_no_ICAO
FROM airtemp_hourly_uk_19852008_no_version0
WHERE id_type != 'ICAO';

'4.) Select records without met_domain_name DLY3208 to new
   table'
SELECT * INTO airtemp_hourly_uk_19852008_no_DLY3208
FROM airtemp_hourly_uk_19852008_no_ICAO
WHERE met_domain_name != 'DLY3208';

'5a.) Count number of observations per day'
SELECT count(ob_date), ob_date, src_id INTO
   count_days_airtemp_no_dly3208
FROM airtemp_hourly_uk_19852008_no_dly3208
GROUP BY ob_date, src_id
ORDER BY ob_date, src_id
'5b.) Remove observations which are part of a day with less
    that 24 hourly observations'
SELECT a.* INTO airtemp_uk_19852008_no_obs_less24per_day
FROM airtemp_hourly_uk_19852008_no_DLY3208 as a,
   count_days_airtemp_no_dly3208 as b
WHERE b.count >= 24 AND a.src_id = b.src_id AND a.ob_date =
    b.ob_date

'6a.) Extract a unique listing of station IDs inside the
   British National Grid'
SELECT src_id, count(src_id) INTO src_id_OSGRID
FROM src_id_list
WHERE grid = 'OS'
GROUP BY src_id;

'6b.) Select observations which only have matching station
   IDs inside the British National Grid'
```

```
SELECT a.* INTO airtemp_hourly_uk_19852008_osgrid_only
FROM airtemp_uk_19852008_no_obs_less24per_day as a,
   src_id_osgrid as b WHERE a.src_id = b.src_id;

'7a.) Count number of hourly observations in British
   National Grid' SELECT count(ob_date), ob_date, src_id
   INTO count_days_airtemp_osgrid_only
FROM airtemp_hourly_uk_19852008_osgrid_only
GROUP BY ob_date, src_id
ORDER BY ob_date, src_id

'7b.) Remove hourly observations still contributing to more
    than 24 hourly observations per day'
SELECT a.* INTO airtemp_uk_19852008_24_obs_per_day
FROM airtemp_hourly_uk_19852008_osgrid_only as a,
   count_days_airtemp_osgrid_only as b
WHERE b.count = 24 AND a.src_id = b.src_id AND a.ob_date =
   b.ob_date
```

Appendix F

A script to find gaps in MIDAS weather station time-series

This Python script uses the datetime module to detect to gaps larger than the specified threshold (in days) in the time-series of MIDAS air temperature measurements. Sequential time-series data are read from standard input (STDIN, e.g. measurements can be selected from the database using the CSQL utility) and the script prints the records before and after and the duration of gaps in the time-series.

F.1 Script source code

```
#!/usr/bin/python
#delta epoch python.
#Check for gaps in time-series data greater than threshold value.
#Tom Holderness 10th October 2009.
import os
import time
import datetime
import sys, string

#read the first value from stdin
line = sys.stdin.readline()
#if not line:
#       break
words = string.split(line,'-')
if len(words) <3:
```

```
        print "Oops! something went wrong. check data input."
        print line
else:
        y0=(words[0])                                   #year0
        m0=(words[1])                                   #month0
        d0=(words[2])                                   #day0
        epoch0=datetime.date(int(y0),int(m0),int(d0))   #first date

#now start loop to read from stdin
while 1:
        line = sys.stdin.readline()     #read first line (line0)
        if not line:                    #stop in EOF
                break
        words = string.split(line,'-')  #split date format
        if len(words) < 3:              #check if year, month, day all present
                break
        else:
                y0=(words[0])                   #year0
                m0=(words[1])                   #month0
                d0=(words[2])                   #day0
                epoch1=datetime.date(int(y0),int(m0),int(d0))   #second date
                threshold=datetime.timedelta(days=1)            #threshold to test
                    against
                #fudge=datetime.date(1901,1,1)
                deltaepoch1=epoch1-epoch0                       #calculate date
                    difference
                if deltaepoch1.days > threshold.days:
                        print '%4s, %4s, %4s' % (deltaepoch1.days -1, epoch0, epoch1)
                elif deltaepoch1.days == 0:
                        null
                        #print "0!", epoch0, epoch1
                elif deltaepoch1.days < threshold.days:
                        print '%4s, %4s, %4s' % (deltaepoch1.days -1, epoch0, epoch1)
                #swap out first date for second
                epoch0=epoch1
```

Appendix G

CSQL - a C programme to extract data from PostgreSQL tables in CSV format

This C programme allows connections to PostgreSQL databases using the libpq library. SQL queries can be executed and the results of a the query are returned to standard output (STDOUT) as comma separated values.

G.1 Source code

```
/*csql.c - connect to a postgresql database in C using libpq
Based on Perkins 2001, "Postgresql", ISBN:1-931841-42-X"
Tom Holderness 09-10-2009
*/

/*
To compile
        cc -I /usr/include/postgresql -lpq  csql_vtest.c -o csql2
To run
        csql 127.0.0.1 MIDAS-SPATIAL postgres password "SELECT ob_date from
            airtemp_19144_19852008_3_4 group by (ob_date) order by (ob_date);"
Notes
        csql2 outputs columns with "," CSV file format.
*/

#include <stdio.h>
```

```
#include <stdlib.h>
#include "libpq-fe.h" /*types and prototypes for libpq*/
//#include <math.h>

/*Prototypes*/
void shutdown(PGconn *conn);

main (int argc, char *argv[], char *env[])
{
        PGconn          *conn; /*Connection to the Database*/
        PGresult        *result;/*result set*/
        int numfields, i, j;

        /*See if there are enough arguments (user validation)*/
        if(argc != 6)
        {
                printf("Usage: csql.exe host dbname username password query\n");
                exit(1);
        }

        conn = PQsetdbLogin(argv[1], "5432", NULL, NULL, argv[2], argv[3], argv[4]);

        /*See if we connected*/
        if(PQstatus(conn) == CONNECTION_BAD)
        {
                printf("Connection to database '%s' failed \n", argv[2]);
                printf("Error was: %s", PQerrorMessage(conn));
                shutdown(conn);
        }

        /*Try a query*/
        result = PQexec(conn, argv[5]);
        if(!result)/*query didn't work*/
        {
                printf("Query Problems\n");
                printf("Error was %s", PQerrorMessage(conn));
                shutdown(conn);
        }

        /*get some descriptive data */
        numfields = PQnfields(result);
        for(i = 0; i< numfields; i++)
        {
                //printf("%-10s",PQfname(result, i));
        }
        //printf("\n");/*Line feed*/
        for(i=0;i<PQntuples(result);i++)
        {
                for(j = 0 ; j<numfields;j++)
                {
```

```
                        printf("%s,",PQgetvalue(result, i, j));
                }
                printf("\n");/*Line Feed*/
        }
        /*Clear out result set*/
        PQclear(result);
        PQfinish(conn);
        exit(0);/*Normal Exit*/
}

/*Close connection and indicate an abnormal exit*/
void shutdown(PGconn *conn)
{
        PQfinish(conn);
        exit(1);
}

/*end*/
```

Appendix H

A Python script to extract and store AVHRR pixel values in a PostGIS table

The Python script 'avhrr2pgdb.py' presented below in Listing H.1 extracts pixel values from a time-series of AVHRR scenes and writes them to a new PostGIS database table using the Psycopg library. The pixels row/column numbers in the script represent the AVHRR pixels at the locations of the selected London weather stations. This script was used to create the spatio-temporally paired data-set of AVHRR EST and MIDAS air temperature measurements for analysis (Section 5.2.2). The script uses the Multiprocessing package is used to parallelise the read process and perform asynchronous writes to the database table. A technical discussion of this software is presented in Section 4.3.

H.1 Script source code

```
#!/usr/bin/env python                                                    1

# avhrr2pgdb - avhrr to postgres. Pull specified pixel values frmo       3
    raster imagery to Postgres.
```

```
# Tom Holderness 01-06-2011

# Script currently set to use two processes using multiprocessing
    module. Each process covers half the file stack and creates its own
     creation to db. This improves speed (no serialization at DB end)
    and avoids connection deadlock. For details see here: initd.org/
    psycopg/docs/usage.html
# As a result currently some database values (destination table) are
    hard coded. Options in place to ba added later if needed.

# Run with
# python avhrr2pgdb.py -d AVHRR_catalogue -o MIDAS-SPATIAL -f /home/
    a5245228/Data/EOBS/AVHRR/Products/TA2/TA2a/


# Change log
# 24-06-2011 - TH - changed output table structure. Now writes one
    temperature per tuple and assigns station IDs based on MIDAS
    weather station src_id (values hard-coded), as a result only one
    est columns is generated. Multiple write (INSERT) statements to
    database are required, not currently multi-threaded.


__version__ = "1.0"

import os, sys, numpy
import psycopg2 # Python-PostgreSQL driver
import psycopg2.extensions
import optparse # Command-line option handler
import pynotify # Notification module
import getpass
from multiprocessing import Process, Queue
from rasterIO import *  # Handle raster IO

def getpixelval(raster, Xpix, Ypix):
        '''Accepts Np raster and returns value of specified pixel'''
        return raster[Ypix,Xpix]

def raster2sql(fdata,database,user,server,password):
```

```
'''Takes list of rasters [pkey,filename], database connection
    and fills target table with required pixel values'''

# Connect to database
conn = psycopg2.connect(database=database, user=user, host=
    server, password=password)
cur = conn.cursor()
for pkey, fname, sat, date, time in fdata:
        # Get raster data
        pointer = opengdalraster(fname)
        band = readrasterband(pointer,1)
        sjp = getpixelval(band,863,878)
        lhr = getpixelval(band,843,881)
        nth = getpixelval(band,845,874)
        kew = getpixelval(band,853,880)
        lwc = getpixelval(band,864,876)
        # Construct SQL
        try:
                # 697 - SJP
                cur.execute("INSERT INTO est_gla_avhrr_t (
                    primary_key, fname, satellite,
                    date_of_scene, time_of_scene, src_id, est)
                    VALUES (%s, %s, %s, %s, %s, %s, %s);",((
                    pkey, fname, sat, date, time, 697, sjp)))
                # 708 - LHR
                cur.execute("INSERT INTO est_gla_avhrr_t (
                    primary_key, fname, satellite,
                    date_of_scene, time_of_scene, src_id, est)
                    VALUES (%s, %s, %s, %s, %s, %s, %s);",((
                    pkey, fname, sat, date, time, 708, lhr)))
                # 709 - NTH
                cur.execute("INSERT INTO est_gla_avhrr_t (
                    primary_key, fname, satellite,
                    date_of_scene, time_of_scene, src_id, est)
                    VALUES (%s, %s, %s, %s, %s, %s, %s);",((
                    pkey, fname, sat, date, time, 709, nth)))

                # 19144 - LHR
                cur.execute("INSERT INTO est_gla_avhrr_t (
                    primary_key, fname, satellite,
```

```
                                date_of_scene, time_of_scene, src_id, est)
                                VALUES (%s, %s, %s, %s, %s, %s, %s);",((
                                pkey, fname, sat, date, time, 19144, lwc)))
                            #print pkey, fname, sjp, lhr, nth, kew, lwc
                                                                                59
                            conn.commit()
                                                                                61
                    except psycopg2.ProgrammingError:
                            sys.stdout.write('Error with file: %s\n' %          63
                                fname)

            cur.close()                                                         65
            conn.close()
                                                                                67
def processRun(q,fdata, database,user,server,password):
            '''Thread queue handler'''                                          69
            # Start queue
            q.put(raster2sql(fdata,database,user,server,password))              71

def main(arg=sys.argv):                                                         73
            '''Main event loop to handle multiple threads'''
                                                                                75
            # Command line options
            d = optparse.OptionParser()                                         77
            d.add_option('--server', '-s', default='127.0.0.1')
            d.add_option('--indatabase','-d',default='postgres')                79
            d.add_option('--outdatabase','-o',default='postgres')
            #d.add_option('--table','-t',default='avhrr2pgdb')                  81
            d.add_option('--user','-u',default='postgres')
            d.add_option('--folder','-f',default='/')                           83
            #d.add_option('--multithread','-m',default='1')
            opts, args = d.parse_args()                                         85

            # Database password                                                 87
            pgs = getpass.getpass('Enter database password: ')
            # Connection to database                                            89
            conn = psycopg2.connect(database=opts.indatabase, user=opts.
                user, host=opts.server, password=pgs)
            # Open database cursor                                              91
            cur = conn.cursor()
```

```
# Get list of files to process from database                    93
SQL = "SELECT t.primary_key, t.fname, o.satellite, o.
    date_of_scene, o.time_of_scene FROM avhrr_ta2a t,
    order_avhrr_1985_2008_cloud0 o WHERE o.primary_key = t.
    primary_key AND o.time_of_scene >= '06:00:00' AND o.
    time_of_scene <= '21:00:00' ORDER BY o.date_of_scene, o.
    time_of_scene;"
cur.execute(SQL)                                                95
fdata = cur.fetchall()
# Close cursor and connection                                   97
cur.close()
conn.close()                                                    99
# Create new empty table to store results
conn = psycopg2.connect(database=opts.outdatabase, user=opts.   101
    user, host=opts.server, password=pgs)
cur = conn.cursor()
                                                                103
# Divide the list into two
length = len(fdata) / 2                                         105

# Idiot check (Tom!)                                            107
raw_input("WARNING! You are about to destroy/create the table '
    est_gla_avhrr_t'. Press Enter to continue.")
# primary_key should be scene_primary_key                       109
SQL = "DROP TABLE IF EXISTS est_gla_avhrr_t; CREATE TABLE
    est_gla_avhrr_t(primary_key numeric(11,0), fname character
    varying(27), satellite character varying(7), date_of_scene
    date, time_of_scene time without time zone, src_id integer,
     est real);"
cur.execute(SQL)                                                111
# Make database changes permanent
conn.commit()                                                   113
cur.close()
conn.close()                                                    115

# Switch Python to working dir before raster2sql reads raster   117
    data
os.chdir(opts.folder)
                                                                119
# fix numpy/psycopg2 conversions
```

```
psycopg2.extensions.register_adapter(numpy.float32, psycopg2.
    _psycopg.AsIs)
## test implementation
## psycopg2.extensions.adapt(numpy.float32(123.456)).getquoted
    ()
# ref: http://stackoverflow.com/questions/5946034/getting-
    recarray-into-postgres-using-psycopg2

# Create a queue to run processes
q = Queue()
p1 = Process(target=processRun,args=(q,fdata[:length],opts.
    outdatabase,opts.user,opts.server,pgs))
p2 = Process(target=processRun,args=(q,fdata[length:],opts.
    outdatabase,opts.user,opts.server,pgs))
# Start processes (dump data to table)
p1.start()
p2.start()
# Wait for process completion before main() can continue
p1.join()
p2.join()
print "sub-processes complete."
# Tidy up the table (primary key etc.)
conn = psycopg2.connect(database=opts.outdatabase, user=opts.
    user, host=opts.server, password=pgs)
cur = conn.cursor()
SQL = "ALTER TABLE est_gla_avhrr_t ADD COLUMN pkey_est_obs
    SERIAL; ALTER TABLE est_gla_avhrr_t ADD PRIMARY KEY (
    pkey_est_obs);"
cur.execute(SQL)
conn.commit()
cur.close()
conn.close()
# Send message to user desktop.
try:
        msg = pynotify.Notification('Python script complete','
            Script "avhrr2.pgdb.py" has completed processing"',
            '/usr/share/icons/gnome/scalable/status/dialog-
            information.svg')
except:
        pass
```

```
        print "script complete."
if __name__ == '__main__':                                              151
        sys.exit(main())
```

Appendix I

Satellite Derived Air Temperature models (SDAT)

I.1 Documentation for the SDAT module

SDAT - Satellite Derived Air Temperature models for the Greater London Authority.

Introduction

SDAT contains a model of air temperature based on regression between AVHRR estimated surface temperature observations and air temperature observations for selected weather stations in the Greater London Authority. It also contains the functionality to create new models based on linear regression between EST and air temperatures.

Selection stations: LWC, LHR, SJP, NTH.

Model form is y = mx+b. Where m is slope and b is y-intercept.

Note that this code has been written with a focus on readability (hence the long variable names). PEP8 style used where possible.

Version: 1.0

Author: Tom Holderness

I.1.1 Functions

runmodel(m, x, b)

Return values of y given line equation for predictor.

model(*surface_temperature*)

Model based on temperatures between May-September 06:00 - 21:00, takes surface temperature and returns predicted air temperature.

buildmodel(*surface_temperature*, *air_temperature*)

Creates model based on linear relationship between surface and air temperatures, returns slope and y-intercept.

mse(*observations*, *predicted*)

Calculate mean squared error of model.

rmse(*observations*, *predicted*)

Calculate root mean squared error of the mode.

crossvalidate(*surface_temperature*, *air_temperature*)

Perform a 2-fold cross validation by comparing a model where 50% of the training data is removed before the model is created and then the model prediction of dependant variable is tested on the remaining (test) data.

Returns MSE of the predicted model

loocv(*surface_temperature*, *air_temperature*)

Performs leave one out cross validation on model derived from supplied data. returns array of y_error values.

Notes:

- x and y must have same length and must be indexed in same order.
- due to the delete function (which creates copy of array, this function is not very efficient.

I.1.2 Variables

Name	Description
__created__	**Value:** 'Tue Jan 24 16:09:58 2012'
__year__	**Value:** '2011'
__package__	**Value:** None

I.1.3 Source code for the SDAT module

```
__version__ = "1.0"

import numpy as np

def runmodel(m, x, b):
    '''Return values of y given line equation for predictor.'''

    return m*x+b

def model(surface_temperature):
    '''Model based on temperatures between May-September 06:00 - 21:00,
    takes surface temperature and returns predicted air temperature.
    '''
    m = 0.571404
    b = 9.918152
    x = surface_temperature

    return m*x+b

def buildmodel(surface_temperature, air_temperature):
    '''Creates model based on linear relationship between surface and
       air temperatures, returns slope and y-intercept.'''

    x = surface_temperature
    y = air_temperature

    z = np.polyfit(x,y,1)

    return z[0], z[1]

def mse(observations, predicted):
    '''Calculate mean squared error of model.'''

    error = observations - predicted
    sq_error = np.power(error, 2)

    return np.mean(sq_error)
```

```
def rmse(observations, predicted):                                    38
    '''Calculate root mean squared error of the mode.'''
                                                                      40
    mserror = mse(observations, predicted)
    return np.sqrt(mserror)                                           42

def crossvalidate(surface_temperature, air_temperature):              44
    '''Perform a 2-fold cross validation by comparing a model where 50%
        of the
    training data is removed before the model is created and then the 46
       model
    prediction of dependant variable is tested on the remaining (test)
       data.
                                                                      48
    Returns MSE of the predicted model'''
                                                                      50
    # Notation:
        # d0 = initial training data (x,y)                            52
        # d1 = observed test data (x,y)
    # Create training and test data sets from surface temperature    54
    np.random.shuffle(surface_temperature)
    #d0x = surface_temperature                                        56
    n = len(surface_temperature)
    i = n/2                                                           58
    d0x = surface_temperature[0:i]
    d1x = surface_temperature[i:n]                                    60

    # Create training and test data sets from air temperature        62
    np.random.shuffle(air_temperature)
    n = len(air_temperature)                                          64
    i = n/2
    d0y = air_temperature[0:i]                                        66
    d1y = air_temperature[i:n]
                                                                      68
    # Get slope and y-intercept for model
    m, b = buildmodel(d0x, d0y)                                       70
    predicted = runmodel(m, d1x, b)
    p1 = rmse(d1y, predicted)                                         72
    # Evaluate quality of prediction using test air dataset.
    print ("Slope␣and␣y-intercept␣of␣crossvalidation␣(pass␣1):␣\      74
```

```
␣␣␣␣␣␣␣␣y␣=␣%.4f␣x␣+␣%.4f" % (m, b))
    print "MSE␣of␣crossvalidation␣(pass␣1):␣%.6f" % mse(d1y, predicted)
    print "RMSE␣of␣crossvalidation␣(pass␣1):␣%.6f" % p1
    print "Number␣of␣obs␣in␣model␣training␣data␣(pass␣1):␣%d" % len(d0y
        )
    print "Number␣of␣obs␣in␣model␣test␣data␣(pass␣1):␣%d\n" % len(d1y)
    # Now repeat but with reversed training/test data (i.e d0/d1).
    m, b = buildmodel(d1x, d1y)
    predicted = runmodel(m, d0x, b)
    p2 = rmse(d0y, predicted)
    pmean = (p1+p2)/2.0
    print type(pmean)
    print ("Slope␣and␣y-intercept␣of␣crossvalidation␣(pass␣2):␣\
␣␣␣␣␣␣␣␣y␣=␣%.4f␣x␣+␣%.4f" % (m, b))
    print "MSE␣of␣crossvalidation␣(pass␣2):␣%.6f" % mse(d0y, predicted)
    print "RMSE␣of␣crossvalidation␣(pass␣2):␣%.6f" % p2
    print "Number␣of␣obs␣in␣model␣training␣data␣(pass␣2):␣%d" % len(d0y
        )
    print "Number␣of␣obs␣in␣model␣test␣data␣(pass␣2):␣%d\n" % len(d1y)

    print "Average␣RMSE␣from␣2-fold␣CV:␣%.6f" % pmean

def loocv(surface_temperature, air_temperature):
    '''Performs leave one out cross validation on model derived from
       supplied
    data. returns array of y_error values.

    Notes:
         - x and y must have same length and must be indexed in same
            order.
         - due to the delete function (which creates copy of array,
            this
             function is not very efficient.'''
    y_error = np.zeros_like(surface_temperature)

    for i in range(0, len(surface_temperature)):
        x_validator = surface_temperature[i]
        y_validator = air_temperature[i]

        x_training = np.delete(surface_temperature, i)
```

```
        y_training = np.delete(air_temperature, i)                    110

        m, b = buildmodel(x_training, y_training)                     112

        y_predicted = runmodel(m, x_validator , b)                    114

        y_error[i] = y_validator - y_predicted                        116

    return y_error                                                    118
```